

1.5em 0pt

THE DIGITAL TWIN CONCEPT AS AN ENABLING TECHNOLOGY IN THE DESIGN OF SYSTEMS



A THESIS SUBMITTED AS PART OF THE REQUIREMENTS FOR THE DEGREE OF
ME MECHANICAL ENGINEERING
AT THE SCHOOL OF MECHANICAL AND MATERIALS ENGINEERING
UNIVERSITY COLLEGE DUBLIN , IRELAND

Author Mark Ryan

Supervisor Dr. Anton Van Beek

Declaration of Authorship

This thesis is the copyright of the student's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Due acknowledgement must always be made of the use of any of the material contained in, or derived from, this thesis.

Signed: Mark Ryan

Date: April 20, 2025

Acknowledgments

I would like to express my sincere gratitude to all those who supported me throughout the course of this thesis.

First and foremost, I am deeply grateful to my supervisor, Dr. Anton Van Beek, for his invaluable guidance, support, and encouragement. His insight and expertise have played a crucial role in shaping the direction and quality of this research, and I have learned a great deal under his mentorship. I hope he continues to inspire others in this fascinating area of research, as he did for me. Another special thanks to Dr. Michael Fop for his advice on the subject.

I would also like to thank my friends, whose constant encouragement and patience made the most challenging parts of this process far more manageable. Your support has been a source of motivation and balance throughout this journey.

Finally, to my family — thank you for your unwavering belief in me. Your love, understanding, and support have been the foundation that allowed me to persevere. This work is as much yours as it is mine.

Abstract

In this thesis, we explore the development and implementation of data-driven, non-parametric predictive models for estimating the degradation and Remaining Useful Life (RUL) of lithium-ion batteries, with a particular contribution being the introduction of an adaptive-grid kernel density estimation (KDE) transition model for improved particle filter performance. It proposes a modification to the standard particle filtering framework to accommodate the challenges of sparse and intermittent sensor data, using Dynamic Bayesian Networks (DBNs) as the modeling structure.

The numerical implementation centers on two main innovations: a modified particle filter capable of incorporating irregular observations, and an adaptive-grid-based KDE transition model to efficiently capture the complex dynamics of battery degradation. The framework is tested on both basic and complex network structures using real-world experimental battery degradation data.

In the basic network, short-term results revealed a lack of belief coherence and weak predictive convergence, especially under sparse update regimes. In contrast, the complex network produced stable posterior distributions and successfully tracked the long-term degradation trend, reflecting an improved representation of hidden battery health. Both models were subject to an effective sample size test to ensure sufficient sample diversity and model validation.

These results demonstrate that model complexity and variable richness play a crucial role in belief evolution, with the complex network offering stronger predictive

power and robustness under practical conditions. Future work could consider exploring hierarchical DBNs and hybrid physics-informed inference models to enhance scalability and support potential integration into digital twin frameworks.

Contents

Declaration of Authorship	ii
Acknowledgments	iii
Abstract	iv
Table of Contents	viii
List of Figures	xii
List of Algorithms	xiii
Nomenclature	

I Thesis

1 Introduction	1
1.1 Overview	1
1.2 Scope of this Thesis	4
2 Literature Review	6
2.1 Bayesian Networks	6
2.1.1 Static Bayesian Networks	6
2.1.2 Hidden Markov Models	8
2.1.3 Dynamic Bayesian Networks	10

2.1.4	Applications of BNs	11
2.2	Bayes' Filters	12
2.2.1	Kalman Filters	13
2.2.2	Particle Filters	13
2.2.3	Transition Models	14
2.3	Kernel Density Estimation	16
2.3.1	Bandwidth Selection	16
2.3.2	Adaptive Gridding	18
2.4	Lithium-Ion Battery Degradation	19
3	Numerical Implementation	21
3.1	Particle Filter	21
3.2	Kernel Density Estimation	22
3.2.1	Bandwidth Selection	22
3.2.2	Limitations of Classical KDE	24
3.2.3	Adaptive Grid KDE	25
3.2.4	KDE Method Comparison	28
4	Basic Network for Battery Degradation	34
4.1	Battery Data Source and Experimental Setup	34
4.2	Proposed Network	35
4.3	Proposed Algorithm I	36
4.4	Results	40
4.4.1	Short Term RUL Indicator	41
4.4.2	Evolution of Belief	42
4.4.3	Long Term RUL Indicator	44
5	Complex Network for Battery Degradation	47
5.1	Proposed Network	47

5.2	Proposed Algorithm II	49
5.3	Results	51
5.3.1	Short Term RUL	51
5.3.2	Evolution of Beliefs	54
5.3.3	Long Term RUL	57
6	Discussion	60
6.1	Validity of Results	60
6.2	Comparison of Networks	63
6.2.1	Short Term RUL Indicators	63
6.2.2	Differences in Evolution of Beliefs	64
6.2.3	Long Term RUL Indicators	64
6.2.4	General Comparison	65
6.3	Future Work	65
6.3.1	Future Applications	65
6.3.2	Transition Models	67
7	Conclusion	70

II Appendices

A Figures	i
Bibliography	v

List of Figures

1.1	Graphical representation of the DT and the digital thread which connects the physical and virtual	4
2.1	Example of a Static BN	7
2.2	HMM example with hidden variable X and observable variable S	9
2.3	Example of a DBN	10
2.4	C. Li et al. 2017 prediction for the crack growth in an aircraft wing along with the prediction intervals obtained through a fracture mechanics transition model.	12
2.5	Lithium ion degradation mechanisms: causes, effects and degradation Signatures. (Birkl 2017)	20
3.1	3D adaptive grid for randomised data. The legend features the corresponding sample spacing for each cluster.	28
3.2	Comparison of methods for 4D artificial data. Slices taken from 3rd and 4th dimension	29
3.3	Comparison of KDE Methods using MATLAB Run Time	30
4.1	The degradation of a lithium ion-battery through the loss in charge capacity coupled with the maximum charging temperature.	36
4.2	The charging temperature profiles of 3 cycles (the 1 st , 4000 th , 7700 th .	37
4.3	The basic DBN for battery degradation prediction featuring two variables	38

4.4	Modified DBN from 4.3 to include the incorporation of intermittent observation data present at timeslice t	39
4.5	Plot showing the actual maximum charge, the mean prediction and the upper and lower 90% prediction intervals.	41
4.6	Plot showing the actual maximum charging temperature, the mean prediction, and the upper and lower 90% prediction intervals.	42
4.7	Ridge plot showing the evolution of the maximum charge distribution across the many charging cycles. Each of these ridges were 25 cycles after a measurement update and show evolution of the distributions. .	43
4.8	Similar to 4.7, the evolution of the maximum charging cycle temperature can be seen with each ridge showing the distribution 25 cycles after a measurement update.	44
4.9	Plot showing the actual maximum charge, the mean prediction and the upper and lower 90% prediction intervals. These are then propagated beyond available data to show the model's prediction and the propagation of uncertainty.	45
4.10	Plot showing the actual maximum charging temperature, the mean prediction and the upper and lower 90% prediction intervals. These are then propagated beyond available data to show the model's prediction and the propagation of uncertainty.	46
5.1	Complex network with two added variables.	48
5.2	(From Left to Right) Maximum charging voltage and the maximum charge, charging time and the maximum charge, maximum charging temperature and the maximum charge. All plotted against the corresponding charge cycle.	48

5.3	Plot showing the actual maximum charge, the mean prediction and the upper and lower 90% prediction intervals from the complex network with four variables.	51
5.4	Plot showing the actual maximum charging temperature, the mean prediction and the upper and lower 90% prediction intervals from the complex network with four variables.	52
5.5	Plot showing the actual maximum voltage, the mean prediction and the upper and lower 90% prediction intervals from the complex network with four variables.	53
5.6	Plot showing the actual charging time, the mean prediction and the upper and lower 90% prediction intervals from the complex network with four variables.	54
5.7	The evolution of the maximum charge from observation to observation and across the full span of the battery life showing the evolution of the belief.	55
5.8	Similar to figure 5.7, the evolution of the maximum charging temperature from observation to observation and across the full span of the battery life showing the evolution of the belief.	55
5.9	The evolution of the maximum charging voltage belief from observation to observation and across the full span of the battery life.	56
5.10	The evolution of the charging time belief from observation to observation and across the full span of the battery life.	56
5.11	Plot showing the actual maximum charge, the mean prediction and the upper and lower 90% prediction intervals. These are then propagated beyond available data to show the model’s prediction and the propagation of uncertainty.	57

5.12 Plot showing the actual maximum charging temperature, the mean prediction and the upper and lower 90% prediction intervals. These are then propagated beyond available data to show the model's prediction and the propagation of uncertainty.	58
5.13 Plot showing the actual maximum charging voltage, the mean prediction and the upper and lower 90% prediction intervals. These are then propagated beyond available data to show the model's prediction and the propagation of uncertainty.	59
5.14 Plot showing the actual charging time, the mean prediction and the upper and lower 90% prediction intervals. These are then propagated beyond available data to show the model's prediction and the propagation of uncertainty.	59
6.1 Plot showing the ESS for the basic network across all timeslices (0 to 7700)	61
6.2 Plot showing the ESS for the complex network across all timeslices (0 to 7700)	62
A.1 Particle paths for charge variable in the basic network.	ii
A.2 Particle paths for maximum charging temperature variable in the basic network.	ii
A.3 Particle paths for charge variable in the complex network.	iii
A.4 Particle paths for maximum charging temperature variable in the complex network.	iii
A.5 Particle paths for maximum charging voltage variable in the complex network.	iv
A.6 Particle paths for charging time variable in the complex network.	iv

List of Algorithms

1	Kalman Filtering Estimation Mechanism	13
2	PF Mechanism	15
3	Modified PF with Intermittent Observation Data	31
4	Leave-One-Out Least Squares Error Cross-Validation Bandwidth Selection	32
5	K-means Clustering Algorithm (Multivariate Data)	33

Nomenclature

Abbreviations

BN	Bayesian Network
CFD	Computational Fluid Dynamics
CPS	Cyber-Physical System
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Network
DT	Digital Twin
FEA	Finite Element Analysis
HMM	Hidden Markov Model
IoT	Internet of Things
ISE	Integrated Square Error
KDE	Kernel Density Estimation
KNN	k-Nearest Neighbour
MCMC	Markov Chain Monte Carlo
MISE	Mean Integrated Square Error
PF	Particle Filter
PDF	Probability Distribution Function
RUL	Remaining Useful Life
SDG	Sustainable Development Goals
SMCM	Sequential Monte Carlo Method

Symbols

A_i	i^{th} Cluster
C	Additional Evidence
C_i	i^{th} Cluster Centroid
K	Kalman Gain Function
K	Kernel
H	KDE Bandwidth (Multivariate Case)
N	Sample Size
O	Observation
Q	State of Charge
R_i	Grid Resolution for i^{th} Cluster
S_i	Sub-Sampling Spacing for i^{th} Cluster
T	Maximum Charging Temperature
V	Internal Voltage
Z	Observation Evidence
c	Smoothing Factor
d	Dimensions
h	KDE Bandwidth (Univariate Case)
n	Samples (KDE Univariate Case)
η	Noise
μ	Sample Mean
ρ_{min}	Minimum Cluster Density
ρ_{max}	Maximum Cluster Density
σ	Covariance
Σ	Covariance Matrix
τ	Charge Time

t Timeslice

w Weight

x Evaluation Point

x_i i^{th} Sample

\mathcal{L} Likelihood

Part I

Thesis

Chapter 1

Introduction

1.1 Overview

Mechanical engineering, as defined by the Massachusetts Institute of Technology, encompasses the responsible development of products, processes, and power, from the nanoscale to the macroscale. It is the broadest of the engineering disciplines, focusing on the design, analysis, manufacturing, and maintenance of mechanical systems by applying principles from mechanics, thermodynamics, materials science, and control theory to real-world problems (*MIT Course Catalog Bulletin 2024-2025* n.d.). The challenges within mechanical engineering may often be summarised as the integration of knowledge from vast sources and various disciplines in the design, analysis, manufacturing, and maintenance of mechanical systems.

Mechanical engineering's relevance to modern global challenges is reflected in its alignment with the United Nations Sustainable Development Goals (SDGs), particularly Goals 9 and 12. Goal 9 emphasizes building resilient infrastructure, promoting inclusive and sustainable industrialization, and fostering innovation. Goal 12 targets responsible consumption and production patterns. Mechanical engineers play a critical role in developing systems and technologies that increase energy efficiency, reduce

waste, and support sustainable industrial growth (Assembly 2015). By integrating advanced technologies and data-driven methods, the field is increasingly centered on creating adaptive, intelligent systems that support both economic and environmental goals.

Industry 4.0 represents a current phase of industrial evolution, characterized by the fusion of Cyber-Physical Systems (CPS), the Internet of Things (IoT), cloud computing, and advanced analytics to create smart, decentralized, and responsive production environments that prioritize system efficiency, quality, adaptability, and continuous improvement. Central to this transformation is the concept of the Digital Twin (DT). NASA defines the Digital Twin as "an integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin" (Glaessgen and Stargel 2012).

The Digital Twin functions as a high-fidelity virtual replica of a physical asset or system, providing a platform for real-time monitoring, simulation, and predictive analysis. By leveraging real-time sensor data, historical usage patterns, and physics-based models, the DT enables improved decision-making, risk mitigation, and optimization of operations. Complementing this is the Digital Thread, a communication framework that connects data flows throughout the product lifecycle, from design and manufacturing to operation and maintenance. While the DT focuses on physics-based modeling and simulation to replicate system behavior in real time, the Digital Thread emphasizes data-driven, machine learning approaches to track, analyze, and optimize system performance across its lifecycle. With the rise of edge computing and ubiquitous IoT connectivity, together, the Digital Twin and Digital Thread enable a closed-loop feedback system that leverages both physical insight and data-driven learning to support predictive maintenance, design optimization, and lifecycle management, and form the backbone of intelligent, adaptive industrial systems,

and hence, Industry 4.0. A graphical representation of this combination closed-loop feedback system that leverages both physical insight and data-driven learning can be seen in figure 1.1.

However, classical modeling techniques face significant limitations when applied to complex, real-world systems. Deterministic models, which rely on well-defined equations describing system dynamics, often struggle to accurately capture the stochastic and non-linear behaviors exhibited by many complex physical systems. Moreover, traditional models typically require complete and evenly spaced datasets—assumptions which are seldom met in practical applications, where data may be sparse and provided through intermittent sensor readings at varying intervals and frequencies.

The economic impact of these shortcomings is substantial. Studies estimate that downtime costs industrial manufacturers up to \$260,000 per hour, with some facilities experiencing hundreds of hours of downtime annually (Arsenault 2016). Such losses are exacerbated by the inability of classical models to accurately predict failures or degradation when data is sparse or corrupted. Predictive maintenance systems that can accurately estimate the Remaining Useful Life (RUL) of components are essential for minimizing downtime and optimizing operational efficiency.

Probabilistic models address some of these limitations by explicitly representing uncertainty in system dynamics and measurements. However, parametric probabilistic models, which assume a fixed functional form and a limited set of parameters, also face challenges when modeling complex systems. Their expressiveness is constrained by predefined distributions and assumptions about the nature of the data, which may not hold in non-linear, high-dimensional, or dynamically evolving systems. These limitations can result in poor generalization, especially when dealing with the sparse, noisy, and irregular data typical in real-world industrial applications.



Figure 1.1: Graphical representation of the DT and the digital thread which connects the physical and virtual

1.2 Scope of this Thesis

This thesis develops and implements a non-parametric¹, data-driven predictive modeling framework for estimating the degradation and Remaining Useful Life (RUL) of lithium-ion batteries. The core contribution is the modification of the standard particle filtering algorithm to handle irregular observations and computational constraints, achieved through two main innovations:

- An adaptive-grid-based KDE transition model, which efficiently captures the complexity and multi-modality of high-dimensional belief spaces by focusing computational effort on dense regions of the state space.
- A modified particle filter capable of incorporating intermittent observation updates, reflecting the reality of many industrial systems where sensor measurements are infrequent, asynchronous, or missing altogether.

These tools are applied to two different Dynamic Bayesian Network (DBN) struc-

¹Non-parametric methods are flexible modeling techniques that do not assume a fixed functional form or architecture, allowing their complexity to adapt based on the available data.

tures, referred to as the “basic” and “complex” networks, each constructed using real experimental battery degradation data. Both networks are evaluated across two key dimensions:

- Short-term performance, assessed via rapid belief updates over closely spaced measurement update intervals.
- Long-term predictive ability, examined through forward propagation well beyond the final observation to assess degradation learning and RUL forecasting.

Rather than the networks being limited to one task or the other, both were assessed under short- and long-term regimes, and the results demonstrate how network complexity affects convergence, belief stability, and predictive accuracy. Particular emphasis is placed on effective sample size (ESS) as a validation metric, showing how model structure and data integration impact particle degeneracy and belief quality over time.

This dual-network evaluation serves not only to validate the methods but also to explore how system complexity, variable selection, and measurement frequency influence the overall reliability and robustness of DBN-based predictive maintenance frameworks.

Chapter 2

Literature Review

2.1 Bayesian Networks

(Kjaerulff 2008) describes Bayesian networks (BN) as directed acyclic graphs (DAG) which defines a factorization of a joint probability distribution over the variables that are represented by the nodes of the DAG, where the factorization is given by the directed links of the DAG. Between the nodes are directed edges which represent causality between variables. They can be particularly useful in modelling systems where a high degree of uncertainty exists, incomplete data exists and the relationships between the variables are complex (S. H. Chen and Pollino 2012).

2.1.1 Static Bayesian Networks

Within static BNs, common structures exist. Most notable V and inverted V structures. Also, terminology exists to describe nodes relative to other nodes. Consider the following example of a BN in figure 2.1.

This BN consists of various classes of nodes or variables. Considering node E as the state/query variable, the following classes of nodes exist.

- Ancestor nodes $\{A, B\}$ (or root nodes) who are parentless and independent.

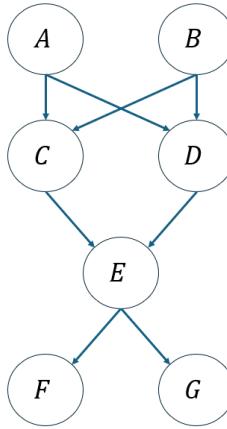


Figure 2.1: Example of a Static BN

- Parent nodes $\{C, D\}$ that depend on the ancestor nodes and that the query node is dependent on.
- Child nodes $\{F, G\}$ that depend on the query node and for this example are observable and will act as evidence.

In order to understand our state variable E , we must use the evidence provided to us by the various nodes within the BN i.e., we want $P(E|Evidence)$ which is the posterior. The Posterior, which refers to our updated belief of a variable or node, is calculated by Baye's theorem which states

$$Posterior = \frac{Likelihood \cdot Prior}{Evidence}, \quad (2.1)$$

where the prior represents the previous belief of the variable and the likelihood is the probability of evidence given the belief.

From the BN, we can see the relationships that exist between the state variable and the parent nodes. So applying Bayes theorem in equation 2.1, one may write

$$P(E|F, G) = \frac{P(F, G|E) \cdot P(E)}{P(G, F)}. \quad (2.2)$$

Since our state variable is itself a child of other variables, and therefore conditional of its parent nodes, one may write

$$P(E) = P(E|D, C) = \frac{P(E, D, C)}{P(D, C)} \quad (2.3)$$

and since the parent nodes are independent of each other, and dependent on the ancestor nodes, one may write equation 2.3 using the probability chain rule.

$$P(E) = \frac{P(E|D, C)P(D)P(C)}{P(D)P(C)}. \quad (2.4)$$

Using the same logic as in equations 2.3 and 2.4, nodes $\{C, D\}$ may be written as

$$P(C) = \frac{P(C|A, B)P(A)P(B)}{P(A)P(B)} \quad \text{and} \quad P(D) = \frac{P(D|A, B)P(A)P(B)}{P(A)P(B)}. \quad (2.5)$$

Combining equations 2.2 and 2.4, one may write the posterior as

$$P(E|F, G) = \frac{P(F, G|E)P(E|D, C)P(D)P(C)}{P(G)P(F)P(D)P(C)}, \quad (2.6)$$

where $P(C)$ and $P(D)$ are obtained from the set of equations 2.5. These fundamental rules and application of Bayes theorem are described in depth by (Kjaerulff 2008).

2.1.2 Hidden Markov Models

HMMs are specific class of DBN and adaptions of Markov chains. They were first introduced by (Baum and Petrie 1966) and feature two variables - a hidden state variable and an observable variable. The variable X_{t+1} is conditional on its previous state X_t , which in turn is only conditional on the previous one. This feature is known as the Markov property (Cappé, Moulines, and Rydén 2005).

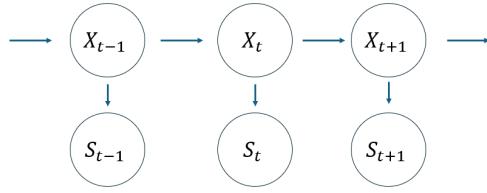


Figure 2.2: HMM example with hidden variable X and observable variable S

(Jurafsky 2000) wrote that a HMM consists of the following components:

- Hidden States: $X = \{X_1, X_2, \dots, X_N\}$ for $t = \{1, 2, \dots, N\}$.
- Observable States: $S = \{S_1, S_2, \dots, S_N\}$ for $t = \{1, 2, \dots, N\}$.
- State Transition Probability Distribution $A = \{a_{ij}\}$ where $a_{ij} = P(X_{t+1}|X_t)$ and $i = t + 1$ and $j = t$ representing the probability of moving from state i to state j .
- Emission Probability Distribution $B = \{b_i\}$ where $b_i = P(S_{t+1}|X_{t+1})$ representing the likelihood of an observation given the state.
- Initial Probability Distribution $\pi = \pi_i = P(X_1)$ representing the probability of the chain starting at state i

The joint distribution of the HMM is obtained by combining transition, emission and initial probability distributions to form a joint distribution given the model parameters in the following following form:

$$P(X, S | A, B, \pi) = \underbrace{P(X_1)}_{\text{initial}} \underbrace{\prod_{i=2}^N P(X_{t+1} | X_t, A, B, \pi)}_{\text{transition}} \underbrace{\prod_{i=1}^N P(S_t | X_t, A, B, \pi)}_{\text{emission}}. \quad (2.7)$$

While HMMs provide a powerful framework for modeling temporal processes with hidden states, their structure is limited to simple, chain-like dependencies between

variables. DBNs build on this foundation by generalizing HMMs to accommodate graphical structures across time slices, enabling richer representations of systems with multiple interacting hidden and observed variables.

2.1.3 Dynamic Bayesian Networks

(Russell and Norvig 2016) defines a dynamic Bayesian network, or DBN, is a Bayesian network that represents a temporal probability model.

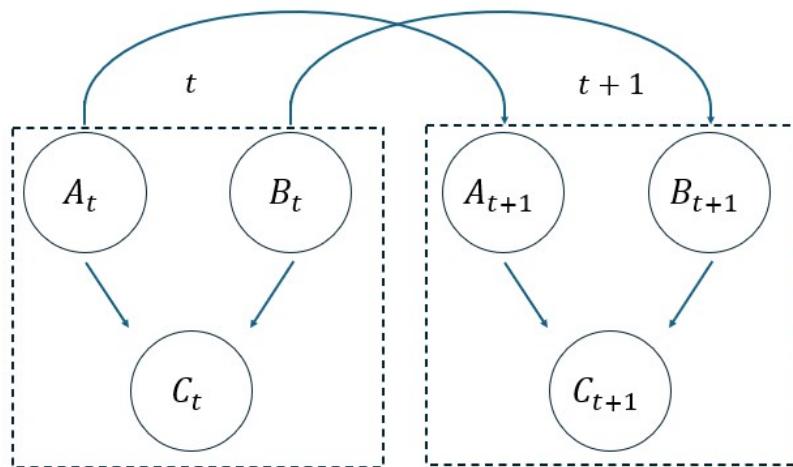


Figure 2.3: Example of a DBN

Considering figure 2.3, one can see a network consisting of three variables or nodes across two time slices. This network consists of two parent variables and one child variable. Considering B_{t+1} as the query node and, C_t and C_{t+1} as observable variables, the DBN can be derived as follows

$$P(B_{t+1}|B_t, C_{t+1}) = P(B_{t+1}|B_t) \cdot P(B_{t+1}|C_{t+1}) \quad (2.8)$$

where, using conditional probability, one obtains

$$P(B_{t+1}|B_t) = \frac{P(B_{t+1}, B_t)}{P(B_t)}, \quad (2.9)$$

and, using Bayes theorem, one obtains

$$P(B_{t+1}|C_{t+1}) = \frac{P(C_{t+1}|B_{t+1}) \cdot P(B_{t+1})}{P(C_{t+1})}. \quad (2.10)$$

This allows for the full probability of B_{t+1} to be written as

$$P(B_{t+1}|B_t, C_{t+1}) = \sum_{B_t, C_{t+1}} P(B_{t+1}, B_t) \cdot P(C_{t+1}|B_{t+1}) \cdot P(B_{t+1}) \quad (2.11)$$

This progression from static BNs to HMMs and ultimately to DBNs illustrates the increasing ability to model temporal structure, uncertainty, and latent state evolution. DBNs retain the interpretability of BNs while introducing the temporal depth of HMMs, making them well-suited for complex, time-evolving systems

2.1.4 Applications of BNs

BNs, both static and dynamic, have found broad and growing application across a variety of domains. Applications extend far beyond engineering with uses ranging from prognosis and diagnosis of diseases as described by (Lucas, Van der Gaag, and Abu-Hanna 2004), but also to ecological surveys as described by (S. H. Chen and Pollino 2012). (Thrun 2002) describes HMM's usage in global positioning systems (GPS) in applications ranging from robotics to satellite navigation in cars to object tracking in factory settings. (Ko and Kim 2012) describes their use in an underwater, autonomous robot.

This thesis is primarily inspired by (C. Li et al. 2017), where a crack growth was monitored in an aircraft wing and a DBN was constructed to track the growth. A PF was used as the method of inference but used a fracture mechanics transition model,

as opposed to this thesis' transition model.

(C. Li et al. 2017) produced a model which both showed the crack prediction along with prediction intervals. Intermittent observations were provided by grounds crew which acted as observation nodes in the DBN. The outcome of their study was a prediction mean as well as prediction intervals which shows the uncertainty quantification obtained by using the DBN and PF method. The results of this study can be seen in figure 2.4.

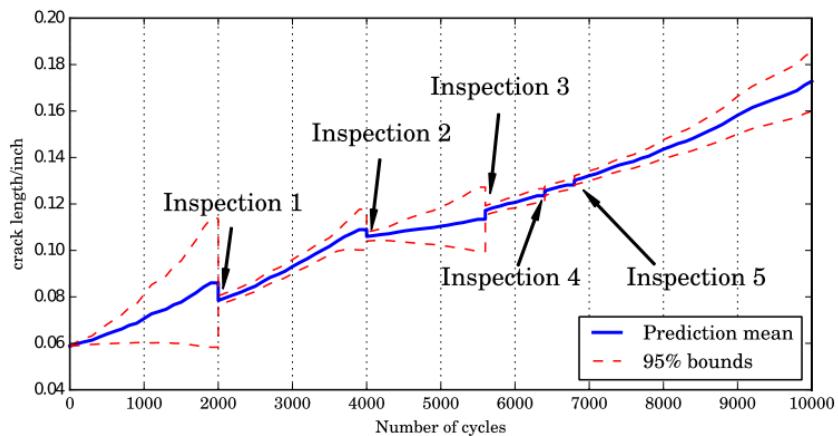


Figure 2.4: C. Li et al. 2017 prediction for the crack growth in an aircraft wing along with the prediction intervals obtained through a fracture mechanics transition model.

2.2 Bayes' Filters

(Särkkä and Svensson 2023) describes Bayes filters as probabilistic algorithms that estimate an unknown probability density function recursively over time. These algorithms compute estimates of the current state of the system given the history of measurements. Applications of Bayesian filtering are described as any phenomena which can be modelled as time varying systems and are common in engineering applications, for example, GPS systems and aircrafts.

2.2.1 Kalman Filters

(Thrun 2002) divides Bayes filters into two families - Gaussian and non-parametric. The most popular of each being Kalman filters and PFs, respectively.

(Kalman 1960) proposed the Kalman filter which presented a recursive solution to the discrete-data linear filtering problem, laying the foundation for modern control theory and applications in various fields. The Kalman filter, is used in linear systems where the system dynamics and observation models are linear and the measurement noise is gaussian. Its main strength stems from the relatively low computational cost compared to other non-parametric methods. The pseudo code for the Kalman filter can be seen in algorithm 1 (Wu et al. 2017).

The algorithm consists of two main steps, step three and four in algorithm 1, which are the prediction step and the measurement update. The Kalman filter uses a Kalman gain function K to determine how much to trust the prediction versus the measurement update (Khodarahmi and Maihami 2023).

Algorithm 1 Kalman Filtering Estimation Mechanism

- 1: **for** $i = 1 : N$ **do**
- 2: **Step 1:** Initialize the related parameters such as F , Q , H , and R .
- 3: **Step 2:** Select the observation value $P_{\text{best}}(t)$.
- 4: **Step 3:** Calculate the estimated value $\hat{X}(t)$ according to the equation:

$$\hat{X}(t) = F \cdot \hat{X}(t-1) + K(t) \cdot (P_{\text{best}}(t) - H \cdot F \cdot \hat{X}(t-1))$$

- 5: **Step 4:** Calculate the predictive value $\hat{X}(t+1)$ using the equation:

$$\hat{X}(t+1) = \text{Normal} \left(F\hat{X}(t), N(t) \right)$$

- 6: **end for**
-

2.2.2 Particle Filters

As opposed to Kalman filters, PFs are non-parametric, non-linear method of inference that uses sequential Monte Carlo methods (SMCM) to represent variables' distribu-

tions. The algorithm can be seen in algorithm 2. The PF consists of four key steps-propagation, reweight, resample and state estimate (this is covered in more detail in section 3.1).

(C. Li et al. 2017) uses a Bayes filter in the form of a PF, a non-parametric filtering method that employs SMCM to estimate uncertainty by calculating the posterior distribution of hidden states in a dynamic system. In this case, the PF is applied within a DBN to quantify uncertainty in the evolution of crack length growth over time based on new observational data. This incorporates the physical model that exists with various sources of aleatoric and epistemic uncertainty ¹ to create a framework for the diagnosis and prognosis of fatigue crack growth in aircrafts with the aim of achieving a robust DT simulation.

(Ko and Kim 2012) compares Kalman and PFs for use in localization of an underwater robot. This paper concludes that the PF results in smoother and more robust trajectory estimation but has a far greater computation time than the Kalman filter.

2.2.3 Transition Models

Key to both Kalman and PFs is the transition model which states how particles are propagated from one state to another in the propogation/prediction stage of the filter. (C. Li et al. 2017) used a fracture mechanics transition model i.e., they used prior knowledge of the system and of fracture mechanics to propagate particles forward.

In contrast, (Darányi, Ruppert, and Abonyi 2024) proposed a method that used kernel density estimation (KDE) as a transition model. More specifically, the test case was tracking a forklift within a warehouse, so the method blended KDE with a state space model to form a transition model. This yielded results that more accurately quantified uncertainty of the forklifts movement and position.

An issue that has often plagued the PF is the issue of sample degeneracy, whereby

¹Aleatoric uncertainty refers to inherent randomness in a system that cannot be reduced, whereas, epistemic uncertainty refers to uncertainty that stems from incomplete knowledge of a system.

Algorithm 2 PF Mechanism

- 1: **Input:** Number of particles N , Initial particles $\{x_0^i\}_{i=1}^N$, Initial weights $\{w_0^i = \frac{1}{N}\}_{i=1}^N$, Observations $\{z_k\}_{k=1}^T$
- 2: **Output:** Estimated state \hat{x}_k
- 3: **for** $k = 1$ to T **do**
- 4: **Step 1: Prediction** - Propagate each particle according to the transition model:
$$x_k^i = f(x_{k-1}^i, u_k) + \eta_k^i, \quad \eta_k^i \sim p(\eta)$$
- 5: **Step 2: Weight Update** - Update the weights of each particle based on the observation z_k :
$$w_k^i \propto w_{k-1}^i \cdot p(z_k | x_k^i)$$
- 6: **Step 3: Normalization** - Normalize the weights:
$$w_k^i = \frac{w_k^i}{\sum_{j=1}^N w_k^j}$$
- 7: **Step 4: Resampling** - Resample particles based on their weights to obtain a new set $\{x_k^i\}_{i=1}^N$ with equal weights:
$$w_k^i = \frac{1}{N} \quad \text{for } i = 1, 2, \dots, N$$
- 8: **Step 5: State Estimation** - Estimate the state at time k as the weighted average of the particles:
$$\hat{x}_k = \sum_{i=1}^N w_k^i x_k^i$$
- 9: **end for**
- 10: **Step 6: Output** - Return the estimated state \hat{x}_k

due to a lack of variance with a variable, the same particle will be repeatedly sampled through the weighted sampling mechanism due to significantly higher probability being associated with a particle. (Musso, Oudjane, and Le Gland 2001) used KDE to form a probability distribution function (PDF) of the particles then was able to smooth the PDF and avoid sample degeneracy. This method used a non-parametric transition model and was able to effectively move through time slices while avoiding particle degeneracy.

2.3 Kernel Density Estimation

KDE is credited to (Rosenblatt 1956), where a set of samples of some variable are used to quantify a variable's PDF. It is a non-parametric method for estimation of a PDF using smoothing kernels, which can be Gaussian or Epanechnikov.

Given a set of univariate samples $X = \{x_1, x_2, \dots, x_n\}$ and an evaluation point on the PDF x where n is the total number of data points in the sample, h is the bandwidth or smoothing factor and K is the Kernel function, the PDF of X may be calculated by using equation 2.12.

$$f(x, h) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (2.12)$$

(Parzen 1962) then expanded KDE to the case of multivariate samples which is written as

$$f(\mathbf{x}, \mathbf{H}) := \frac{1}{n|\mathbf{H}|^{1/2}} \sum_{i=1}^n K\left(\mathbf{H}^{-1/2}(\mathbf{x} - \mathbf{X}_i)\right), \quad (2.13)$$

where \mathbf{H} is the bandwidth matrix and \mathbf{X} consists of n samples and d dimensions.

2.3.1 Bandwidth Selection

(Chacón and Duong 2018) describes the bandwidth h or, in the multivariate case, H is a free parameter which effectively determines the smoothness of a distribution. The bandwidth of a KDE has far more of an impact on the density estimate than the kernel selected as detailed by (Turlach 1999).

In the multivariate case, H is defined as

$$\mathbf{H} = \Sigma \cdot c, \quad (2.14)$$

where Σ is the covariance matrix and c smoothing factor.

The selection of bandwidth may be determined through a variety of methods but can be broken into three main classes. (Chacón and Duong 2018) refers to these classes as “quick and dirty” methods, cross-validation methods, and plug-in methods.

The most famous “quick and dirty” method is Silverman’s Rule of Thumb. (Silverman 1986) proposed the following method for bandwidth selection, dependent on sample size and sample dimensions. The main issue with this method is the assumption of unimodality, however, it correctly increases smoothness with dimensions due to the “curse of dimensionality”, as stated by (Bellman, Kalaba, et al. 1965). With this in mind, one can see how the main influence on this rule of thumb is smoothness compensating for data sparsity as seen in equation 2.15, where a dataset’s, of n samples and d dimensions, bandwidth is calculated.

$$\mathbf{H} = \left[\left(\frac{4}{d+2} \right)^{\frac{2}{d+4}} n^{-\frac{2}{d+4}} \right] \cdot \Sigma \quad (2.15)$$

More sophisticated methods exist in the other classes of methods. The most known cross-validation method is the Least Squares method proposed by (Bowman 1984) and (Rudemo 1982). Within this method stems others, mainly around the choice of loss function which are most commonly L_2 functions. These methods work by measuring the error between the density estimate and the true density function.

Since the true underlying function is unknown, the leave-one-out technique provides a practical approach to evaluating bandwidth selection. This method involves constructing the full KDE using a candidate bandwidth, and then systematically recomputing the KDE by leaving out one sample at a time. The error is calculated as the difference between the full KDE and the leave-one-out KDE for each omitted point. By repeating this process for all samples and summing the resulting errors, a cumulative measure of bandwidth error is obtained. This approach effectively evaluates how well the KDE generalizes to unseen data points, providing a robust criterion for bandwidth selection.

The plug-in method for bandwidth selection KDE was formalized by (Jones and Sheather 1991) and by (Hall et al. 1991). This approach aims to minimize the Mean Integrated Squared Error (MISE) by plugging in estimates of unknown quantities, such as the second derivative of the density function, into an asymptotic formula for optimal bandwidth selection. Unlike cross-validation methods, which directly evaluate the accuracy of a KDE, plug-in methods estimate the underlying density's smoothness through pilot estimates. This makes them computationally efficient but potentially sensitive to the choice of pilot bandwidth, particularly when the true density is multimodal or heavy-tailed.

2.3.2 Adaptive Gridding

As stated by (Bellman, Kalaba, et al. 1965), the curse of dimensionality is a real issue with KDE as a method. Due to the computational complexity of KDE being $\mathcal{O}(n \cdot k^d)$ where k is the number of evaluation points per dimension.

The idea of adaptive gridding is nothing new in the context of KDE, however, the majority of the work has been focused on fitting local bandwidths to subregions of the global grids, such as the work done by (Simon Sheather 1983) and (Simon J Sheather 1986). These improved bandwidth selection techniques improve the accuracy of density estimation by avoiding fitting a single global bandwidth. In a similar fashion, (Lemke et al. 2015) used local kernels as opposed to a global kernel. They did grouping samples using k-nearest neighbour (KNN), then deriving a local kernel using a multivariate Gaussian distribution and a local covariance for that specific cluster.

Improving the efficiency of KDE is an explored space but with various distinct approaches to it. (Zhang et al. 2023) combats the high computational cost associated with performing KDE on large datasets by transforming the data distribution to uniform distribution, then, applying a space-filling design and finally mapping the

design points back to the original space. This results a structured subsample that effectively represents the larger dataset.

The idea of adaptive evaluation grids used in KDE is ripe for exploration in order to increase the efficiency of the method for high dimensional data. These evaluation grids would be similar to the adaptive grids one sees in CFD and FEA simulations. Similar approaches are used in noisy Markov chain Monte Carlo (MCMC), like that detailed in (Rastelli, Maire, and Friel 2024), whereby a noisy MCMC approach is used for fitting latent position models. This paper details overlaying a grid across the latent space and using the box centres that form the grid as representative points of the full dataset, and then calculate likelihood based on the box centres.

2.4 Lithium-Ion Battery Degradation

Lithium-ion batteries possess a major advantage over other battery technologies due to their high specific energy density and volumetric energy density (D. Chen 2012). These properties have made them the leading battery technology for use in mobile devices since the 1990s. More recently, they are widely used in electric vehicles and as energy storage solutions, with the The global lithium-ion battery market size was valued at USD 59.8 Billion in 2022 and is projected to reach USD 307.8 billion by 2032 (Market.US 2023).

Despite their numerous advantages over other battery technologies, their performance and safety are often compromised by degradation mechanisms and failures which primarily dependent on conditions such as state of charge, depth of discharge, and temperature (Menye, Camara, and Dakyo 2025).

The degradation causes, mechanisms and effect are detailed in figure 2.5 supplied by (Birkl 2017). The degradation process is important to track but extremely difficult to predict due to the complex interaction between the cause variables such as state

of charge, voltage, temperature and time of the charge and discharge cycles. The degradation process is not just important for RUL determination but also safety of the user. (Ohsaki et al. 2005) found that safety hazards can arise from lithium plating and dendritic growth on graphite anodes, which may cause internal short-circuits, potentially resulting in fires and/or explosions. Similarly, (Monroe and Newman 2003) found that safety hazards can arise from lithium plating and dendritic growth on graphite anodes, which may cause internal short-circuits, potentially resulting in fires and/or explosions.

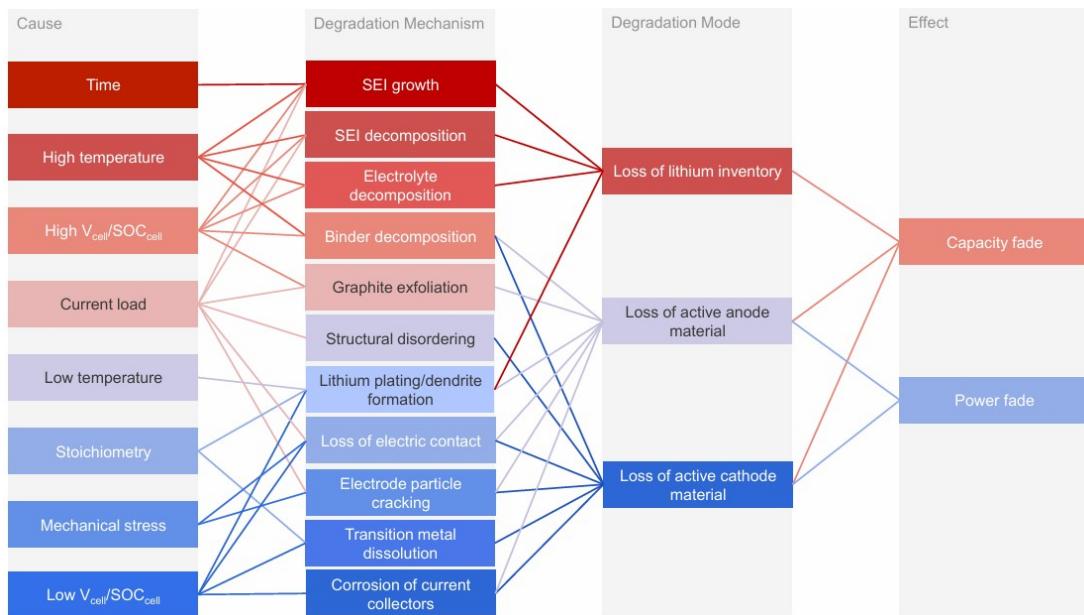


Figure 2.5: Lithium ion degradation mechanisms: causes, effects and degradation signatures. (Birk 2017)

Chapter 3

Numerical Implementation

This chapter is devoted to the explanation and justification of the numerical methods and algorithms implemented in this thesis.

3.1 Particle Filter

As discussed in section 2.2.2, a PF is a non-linear, non-parametric method of inference that uses SMCM to represent variables' distributions. The standard PF, found in applications such as GPS, updates a distribution only when a measurement becomes available. In this thesis, the PF completes the prediction stage at fixed intervals despite whether a measurement is available or not. If a measurement is available, the prediction, reweight, resample and state estimate are completed. With this in mind, consider the updated algorithm 3.

Note the inclusion of the if statement and the general inclusion of the evidence input. This forms the backbone of the PF algorithm, whereby, now particles are proposed to form a prior and the evidence is then considered to form a pseudo posterior either via the conditional probability rule or by Baye's theorem depending on the BN or DBN structure. Particles are proposed stochastically but then form a prior, to then form a pseudo posterior via KDE, which is then sampled to generate particles that

represent the posterior rather than the prior. From then, if an observation variable is available, steps two, three and four are carried out. If not, step five is completed with the posterior particles from step one.

The evidence to be considered to form the pseudo posterior would be particles also subject to the PF algorithm with their prior distribution being considered with the Markov property, and, if available, other evidence, also depending on the structure of the BN or DBN.

3.2 Kernel Density Estimation

In this thesis, KDE forms the transition model and acts as an intermediate mechanism to create distributions from particles in the same way, sampling creates particles from distributions.

3.2.1 Bandwidth Selection

As discussed in the section 2.3.1, various methods for bandwidth selection exist. This thesis looked at both Silverman's rule of thumb, as a benchmark and a leave-one-out cross validation method for a more accurate bandwidth calculation.

However, firstly, the method of calculation of a sample covariance matrix Σ for a dataset $\{\mathbf{X}_i\}_{i=1}^n$ where each $\mathbf{X}_i \in \mathbb{R}^d$, is a $d \times d$ matrix with entries:

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1d} \\ \Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{d1} & \Sigma_{d2} & \cdots & \Sigma_{dd} \end{bmatrix} \quad (3.1)$$

Each entry Σ_{jk} is calculated as:

$$\Sigma_{jk} = \frac{1}{n-1} \sum_{i=1}^n (X_{ij} - \mu_j)(X_{ik} - \mu_k) \quad (3.2)$$

where:

$$\mu_j = \frac{1}{n} \sum_{i=1}^n X_{ij} \quad (3.3)$$

- X_{ij} is the j -th component of the i -th data point \mathbf{X}_i . - μ_j is the mean of the j -th component across all samples.

The bandwidth matrix is then simply

$$\mathbf{H} = \Sigma \cdot c \quad (3.4)$$

where c is the smoothing factor calculated through the methods outlined in the following sections.

Silverman's Bandwidth Selection

Silverman's rule of thumb for estimating the optimal bandwidth matrix \mathbf{H} in a d -dimensional space with n samples is given by:

$$\mathbf{H} = \left[\left(\frac{4}{d+2} \right)^{\frac{2}{d+4}} n^{-\frac{2}{d+4}} \right] \cdot \Sigma \quad (3.5)$$

This rule of thumb served as a benchmark bandwidth calculation due to the simplicity of this rule of thumb, and comparisons were made with this bandwidth calculation method when evaluating the performance of other methods. As discussed in section 2.3.1, this method smooths according to data sparsity that occurs with increased dimensions and decreased samples.

Cross Validation Bandwidth Selection

The method of cross validation bandwidth selection utilizes loss functions in the calculation of optimal bandwidths. As outlined in algorithm 4, the cross validation method used in this thesis is a “leave-one-out” least squares method. The choice of L_2 loss function was integrated square error (ISE), which is a better fit over other L_2 functions such as MISE due to its ability to give an exact measure of the estimator’s performance for a given sample as opposed to the more theoretical MISE which focuses on aiding understanding over whole regions rather than specific samples.

The full method works by first proposing candidate bandwidths. For that bandwidth a full KDE is evaluated and then, the method loops through each sample where that sample is left out. The KDE is then evaluated at all but that sample and the error between the sample and the full KDE at that point is calculated. Looping through all samples, the sum of the error for that bandwidth is calculated and the ISE of that bandwidth is calculated. Looping through all candidate bandwidths, the optimal bandwidth will correspond with the least ISE and that bandwidth will be used for the KDE evaluation. For this thesis, the ISE was calculated using the trapezoidal rule for a approximation of the integration.

The obvious limitations of this method is the computational complexity relative to that of Silverman’s Rule of thumb. The computational complexity for this method is approximately $\mathcal{O}(kmn^2)$ where k is the number of candidate bandwidths, m is the number of evaluation points and n is the number of samples.

3.2.2 Limitations of Classical KDE

As mentioned in section 2.3.2, the computational complexity of KDE, without an elaborate bandwidth selection process, is $\mathcal{O}(nm)$. As seen in the previous section, the incorporation of a cross validation method introduces a far higher complexity.

The “curse of dimensionality” holds true when scaling the evaluation points with

dimensions as $m_{total} = m^d$ where m is the evaluation points in one dimension and d is the number of sample dimensions. While reducing grid size offers the obvious solution to this, the loss in resolution can be detrimental to the model accuracy.

3.2.3 Adaptive Grid KDE

Section 2.3.2 discusses the previous work that relates to adaptive gridding, namely, local bandwidth and local kernels. A key contribution of this thesis is the introduction of a novel KDE method that leverages locally adaptive evaluation point densities — a concept previously unexplored in the context of particle-based inference, as the use of KDE as a transition model remains a relatively unexplored area. This approach was specifically developed to improve computational efficiency and resolution in high-density regions of the belief space, and forms a central part of the methodological advancements presented in this thesis.

Overview

The classical KDE uses regular grids that range in dimension. Although, the density of grids in different dimensions may be varied, how might one try to be more sophisticated in their approach. This thesis proposes a method of first clustering samples into regions of high density and low density. Using a regular global grid, the volume of each cluster can be estimated. Then using a set minimum and maximum density, a sub-sampling space can be calculated for each cluster. Then using this spacing, each cluster has sets of regular evaluation points but globally are irregular. Then evaluating KDE at these points, the PDF will be more accurate in sample rich, high probability zones, and less so in the sparse zones. Considering the application of this method as being sampled from, this loss in accuracy in sparse zones will not have a large effect on the particles created.

This method differs from that proposed by (Zhang et al. 2023), whereby, the

number of evaluation points are reduced rather than the samples. Application of the method proposed by (Zhang et al. 2023) may result in sample degeneracy in a PF and would likely not be fit for purpose in this particular application.

K-Means Clustering

Clustering is a common tool in data science and is defined as an unsupervised approach to machine learning (Sinaga and Yang 2020). Many variations of clustering algorithms exist but the main two families are probability model based and non-parametric approaches. The most popular and oldest method is the partitional k-means clustering method, where data is grouped into clusters by minimizing an objective function. This objective function essentially define dissimilarity between data points.

The method imposed in this paper uses the partitional k-means clustering method with a Euclidean distance objective function as seen in algorithm 5. Once points have been assigned to their nearest cluster, the centroids of the clusters are recalculated and the method is repeated until the cluster centroids converge.

Within the wider adaptive grid KDE method, this is done first to evaluate regions of high density. Once clusters have been made, the global evaluation points can be assigned to each cluster.

Sub-Sampling

Given the partitioned global space, one wishes to now set the number of evaluation points in each cluster to be representative of the sample density. So for large clusters, the evaluation point density should be lower than that of the high density regions.

To achieve this, a volume estimate has to be made using the number of global evaluation points within each cluster. Let N be the total number of global evaluation points, k be the total number of clusters, C_i be the i^{th} cluster and n_i be the number

of global evaluation points within the i^{th} cluster, the proportionate volume of each cluster may be calculated as

$$A_i = \frac{n_i}{N}. \quad (3.6)$$

If the user then prescribes a minimum cluster density, ρ_{min} , and a maximum cluster density ρ_{max} , the grid resolution for each cluster can be calculated as

$$R_i = \rho_{min} + (\rho_{max} - \rho_{min})(1 - A_i). \quad (3.7)$$

To achieve the varying grid spacing for both the results in figures 3.1 and 3.2, ρ_{min} was set to 50 particles and ρ_{max} was set to 2000 across a $40 \times 40 \times 40 \times 40$ grid.

Then the spacing for the sub-sampling in each cluster can be calculated as

$$S_i = round\left(\frac{n_i}{R_i}\right). \quad (3.8)$$

Where the points to be evaluated in the KDE are $x = \{x_j \in C_i | j = 1, S_i, 2S_i, \dots\}$. The adaptive grid produced results in a new grid of points at more sophisticated positions which maximize useful information gain and reduce needless computational expenditure in sparse regions. A 3D representation of this can be in figure 3.1. The irregularity of the new evaluation points can be seen which requires interpolation back onto a regular grid for sampling.

Interpolation

Once the KDE has been evaluated at the irregular evaluation points, the PDF obtained needs to be interpolated back onto a regular grid for sampling in the PF. Once again, k-means clustering is used to identify the closest PDF points to the grid points. The number of points to interpolate from acts as an input and the order of interpolation also. Once interpolated to the global grid, one has the alternative to

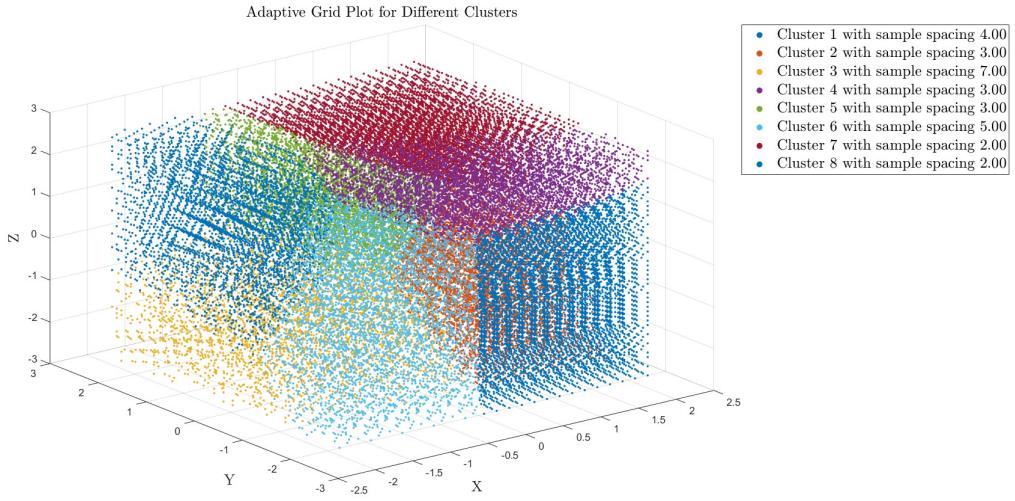


Figure 3.1: 3D adaptive grid for randomised data. The legend features the corresponding sample spacing for each cluster.

the standard evaluation point KDE method.

3.2.4 KDE Method Comparison

To contrast the differences in these methods, artificially randomised data was produced. The data produced followed a Gaussian distribution and consisted of four dimensions and forty data-points. These comparisons can be seen graphically in figure 3.2 with inputs seen in table 3.1.

Table 3.1: KDE Adaptive Grid Testing Parameters

Parameter	Value
ρ_{min}	50
ρ_{max}	2000
Particles	40

The first difference one notices is with the adaptive grid more grainy than the others. The reason for the graininess is the interpolation back onto the regular grid as opposed to the other two methods where a far smoother surface is obtained. Despite this, one gets a relatively accurate picture of the PDF. It's also worth noting,

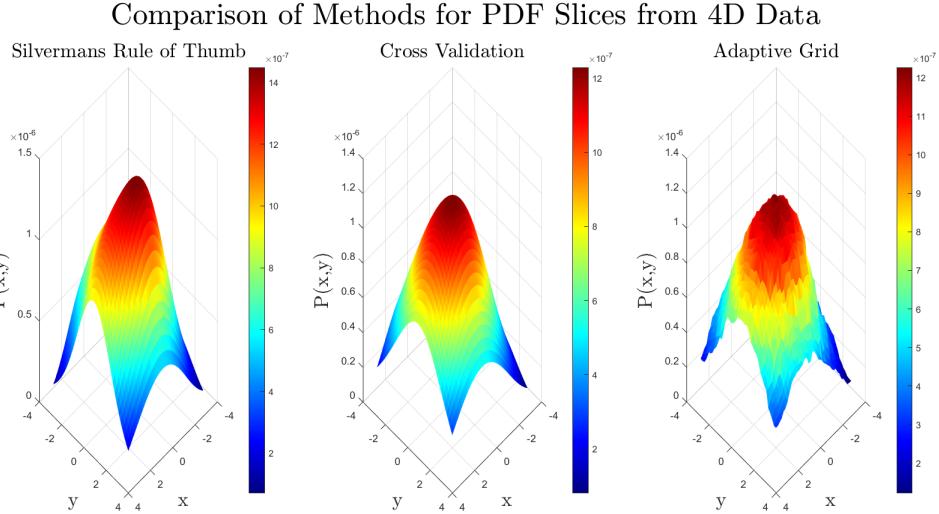


Figure 3.2: Comparison of methods for 4D artificial data. Slices taken from 3rd and 4th dimension

the capturing of the multimodality in the dense central region with characteristic spikes. Due to the main function of these PDFs to eventually be sampled from, the multimodality in these regions is actually an advantage for getting variety amongst the samples. Furthermore, the relative smoothness of the four low probability zones indicates that the bandwidth selection was correct for this PDF. The bandwidth for the adaptive grid was achieved using the “leave-one-out” cross validation bandwidth selection.

Comparing the first and second surfaces, a slight skew can be seen in the Silverman’s rule of thumb surface as opposed to the cross-validation one. Due to the relative sparsity of the data, Silverman’s rule still tries to oversmooth the PDF but clearly it does not smooth it enough as the Gaussian surface is far closer to that of the “leave-one-out” cross validation bandwidth selection.

Figure 3.3 compares the relative speed at which MATLAB computes the PDF for varying sample sizes and grid sizes. The speed tests were ran with the following datasets in table 3.2.

Naturally, the evaluated KDE for the adaptive grid would be smaller than that

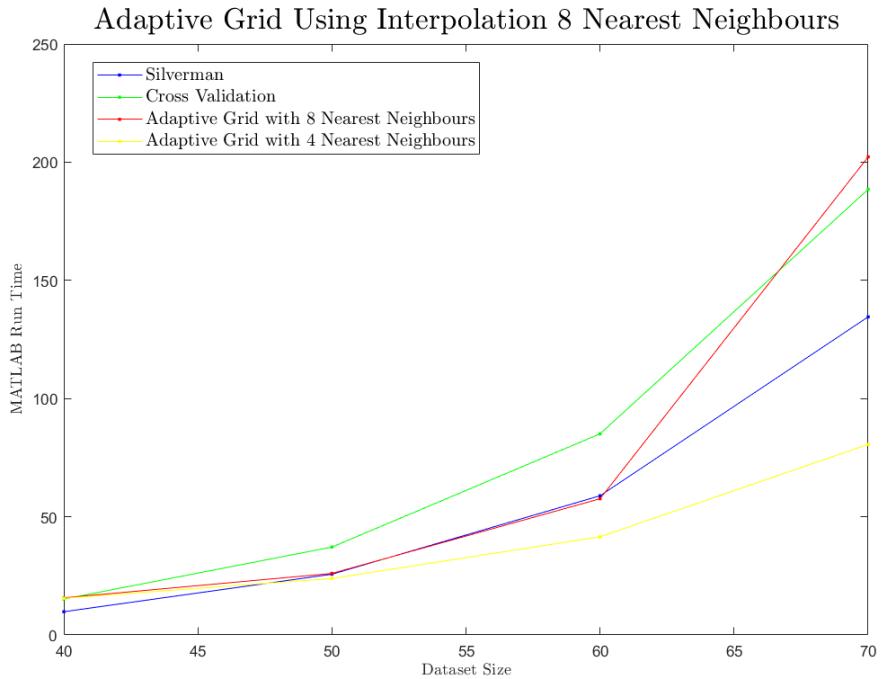


Figure 3.3: Comparison of KDE Methods using MATLAB Run Time

Table 3.2: Sample Size vs. Global Grid Size

Sample Size (N)	Global Grid Size (N^d)
40	2,560,000
50	6,250,000
60	12,960,000
70	24,010,000

of the global grid size. The adaptive grid was also ran twice for each case with eight nearest neighbours for the interpolation and four nearest neighbours. The difference between both in figure 3.3 is substantially large with the larger grid size, owing to the computational cost associated with the KNN algorithm. The far better performance of the lower KNN value for the adaptive grid shows its far better performance with high dimensional data.

Algorithm 3 Modified PF with Intermittent Observation Data

- 1: **Input:** Number of particles N , Initial particles $\{x_0^i\}_{i=1}^N$, Initial weights $\{w_0^i = \frac{1}{N}\}_{i=1}^N$, Observations $\{z_k\}_{k=1}^T$, Additional Evidence $\{C_k\}_{k=1}^T$
- 2: **Output:** Estimated state \hat{x}_k
- 3: **for** $k = 1$ to T **do**
- 4: **Step 1: Prior Formation with Evidence** - Generate particles by propagating from the previous state, incorporating evidence from other variables:

$$x_k^i = f(x_{k-1}^i, u_k, C_k) + \eta_k^i, \quad \eta_k^i \sim P(\eta)$$

These particles form a pseudo-posterior distribution:

$$P(x_k \mid x_{k-1}, u_k, C_k)$$

- 5: **if** Observation z_k is available **then**
- 6: **Step 2: Posterior Formation** - Update the weights of each particle using the observation z_k :

$$w_k^i \propto w_{k-1}^i \cdot P(z_k \mid x_k^i)$$

The updated particles and weights now represent the posterior distribution:

$$P(x_k \mid z_{1:k}, C_{1:k})$$

- 7: **Step 3: Normalization** - Normalize the weights:

$$w_k^i = \frac{w_k^i}{\sum_{j=1}^N w_k^j}$$

- 8: **Step 4: Resampling** - Resample particles based on their weights to form a new set $\{x_k^i\}_{i=1}^N$ with equal weights:

$$w_k^i = \frac{1}{N} \quad \text{for } i = 1, 2, \dots, N$$

- 9: **else**
- 10: **No Observation Available:** Proceed to Step 5 using the pseudo-posterior particles from Step 1.
- 11: **end if**
- 12: **Step 5: State Estimation** - Estimate the state at time k as the weighted average of the particles:

$$\hat{x}_k = \sum_{i=1}^N w_k^i x_k^i$$

- 13: **end for**
 - 14: **Step 6: Output** - Return the estimated state \hat{x}_k
-

Algorithm 4 Leave-One-Out Least Squares Error Cross-Validation Bandwidth Selection

1: **Input:** Data samples $\{\mathbf{X}_j\}_{j=1}^n$, Candidate bandwidths $\{H_i\}_{i=1}^m$
2: **Output:** Optimal bandwidth H^*
3: **for** $i = 1$ to m **do**
4: **Step 1:** Initialize $LS(H_i) \leftarrow 0$
5: **Step 2:** Compute the KDE $f(\mathbf{X}; H_i)$ using all samples
6: **Step 3:** Initialize Leave-One-Out Sum to 0: $\sum f_{-k}(\mathbf{X}_k; H_i) \leftarrow 0$
7: **for** $j = 1$ to n **do**
8: **Step 4:** Exclude sample \mathbf{X}_j and compute $f_{-j}(\mathbf{X}_j; H_i)$
9: **Step 5:** Accumulate:

$$\sum_{k=1}^j f_{-k}(\mathbf{X}_k; H_i) = \sum_{k=1}^{j-1} f_{-k}(\mathbf{X}_k; H_i) + f_{-j}(\mathbf{X}_j; H_i)$$

10: **end for**
11: **Step 6:** Compute $I(H_i)$ via numerical integration:

$$I(H_i) = \int f^2(\mathbf{X}; H_i) d\mathbf{x}$$

12: **Step 7:** Compute the Least Squares Error:

$$LS(H_i) = I(H_i) - \frac{2}{n} \sum_{j=1}^n f_{-j}(\mathbf{X}_j; H_i)$$

13: **end for**
14: **Step 8:** Select optimal bandwidth:

$$H^* = \arg \min_{H_i} LS(H_i)$$

15: **Step 9: Return** H^*

Algorithm 5 K-means Clustering Algorithm (Multivariate Data)

Require: Dataset $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d$, Number of clusters K , Maximum iterations T

Ensure: Cluster assignments C_i and centroids $\mu_1, \mu_2, \dots, \mu_K$

1: Initialize centroids $\mu_1, \mu_2, \dots, \mu_K$ randomly from X

2: Initialize $t \leftarrow 0$

3: **while** $t < T$ **do**

4: **Assignment Step:**

5: **for** each data point $\mathbf{x}_i \in X$ **do**

6: Assign \mathbf{x}_i to the nearest centroid:

$$C_i = \arg \min_{j \in \{1, \dots, K\}} \|\mathbf{x}_i - \mu_j\|^2$$

7: **end for**

8: **Update Step:**

9: **for** each cluster $j = 1, \dots, K$ **do**

10: Recalculate centroid μ_j :

$$\mu_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$$

11: **end for**

12: **Convergence Check:**

13: **if** Centroids have stabilized or $t = T$ **then**

14: Break

15: **end if**

16: $t \leftarrow t + 1$

17: **end while**

18: **Output:** Cluster assignments C_i and centroids $\mu_1, \mu_2, \dots, \mu_K$

Chapter 4

Basic Network for Battery Degradation

4.1 Battery Data Source and Experimental Setup

The test data that was used for this thesis was compiled by (Birkl 2017) as part of his PhD thesis. Long term cycling tests conducted on commercial Kokam 740 mAh pouch cells were subjected to controlled driving profiles and periodic diagnostic cycles to simulate realistic electric vehicle usage conditions. Each cell was exposed to repeated Artemis urban drive cycles, which emulate stop-start driving patterns with alternating current loads and regenerative braking. The test was conducted at an elevated temperature of 40°C to accelerate degradation, with charging performed at 2°C (1480 mA) and characterisation cycles performed every 100 drive cycles using low-rate charge and discharge procedures (C/18.5, i.e., 40 mA). Voltage, current, and temperature measurements were recorded at regular intervals throughout the test.

The cells were housed in a Binder thermal chamber to ensure a controlled environment. All experiments were carried out using a Bio-Logic MPG-205 battery tester, and the cells were mounted on custom holders to ensure stable electrical contacts over

long durations.

These rigorous and repeatable experimental conditions provided a well-labeled dataset suitable for training and DBNs for degradation modeling and RUL prediction.

4.2 Proposed Network

The basic network used a DBN consisting of two nodes - maximum charging cycle temperature and maximum state of charge, using the maximum state of charge as the indicator of the battery health and therefore, the RUL of the battery. Meanwhile, the maximum charging temperature would act as a parent node, therefore, a variable on which the RUL was dependent on. (Z. Li et al. 2011) created a life cycle model that included temperature as a stress coupling factor and concluded it had a non-negligible impact on the RUL of a battery.

If one considers figure 4.1, the charge degradation with the corresponding maximum charging temperature can be seen across the cycles. It should be noted the lack of clear relationship between both variables. This hints at the need for a non-parametric inference method to be used to describe this relationship.

Coupling this with figure 4.2, the maximum temperature can be seen to be a relatively strong indicator of the degradation of the battery owing to the high starting temperature in cycle 7700 versus with cycle 1.

For that reason, the proposed basic network looks like figure 4.3, where the DBN is an altered Markov chain where the maximum charging temperature informs the maximum state of charge and that then informs the next maximum charging temperature at timeslice $t + 1$. Then, upon an intermittent measurement providing data for observation nodes, the network would transform to that of figure 4.4.

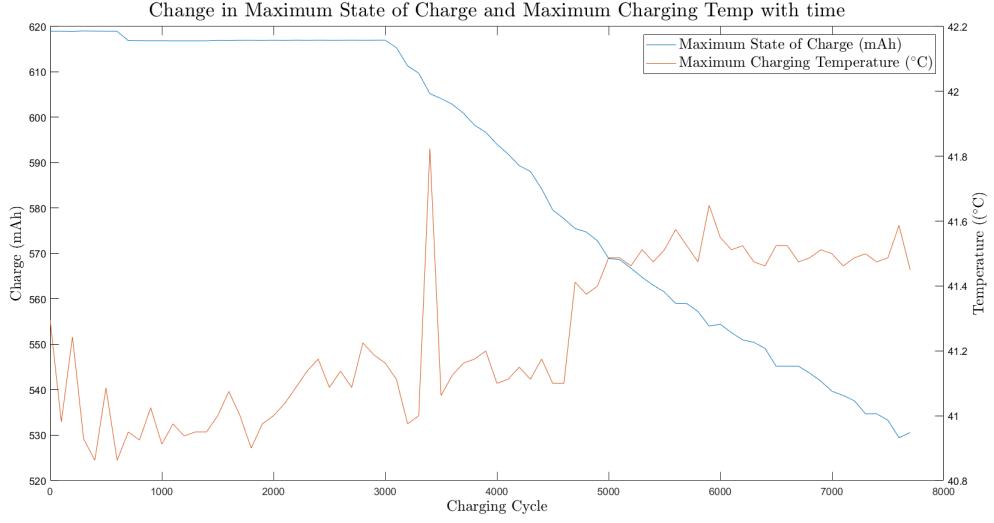


Figure 4.1: The degradation of a lithium ion-battery through the loss in charge capacity coupled with the maximum charging temperature.

4.3 Proposed Algorithm I

Building off of the proposed the network, the objectives of the algorithm become clear. Firstly, initializing using the charging data provided at the first cycle to set particles to incorporate the observation noise i.e. using the Gaussian likelihood formula intialize a posterior. Then, the particles are propagated stochastically until the next observation at cycle 100, where the observation is accounted for also using the Gaussian likelihood formula as seen in equation 4.1 where $\mathcal{L}(x_i | \theta, \sigma)$ is the likelihood of a value given both the model output $f(\theta)$ and observation noise σ .

$$\mathcal{L}(x_i | \theta, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i - f(\theta))^2}{2\sigma^2}\right). \quad (4.1)$$

In terms of probability, the probability equations for cycles without observations like in figure 4.3 would appear by defining priors and then using conditional probability to define posteriors. Let the maximum charging temperature at time t as T_t and the maximum state of charge as Q_t and likewise for the next timeslice $t + 1$ as T_{t+1} and Q_{t+1} .

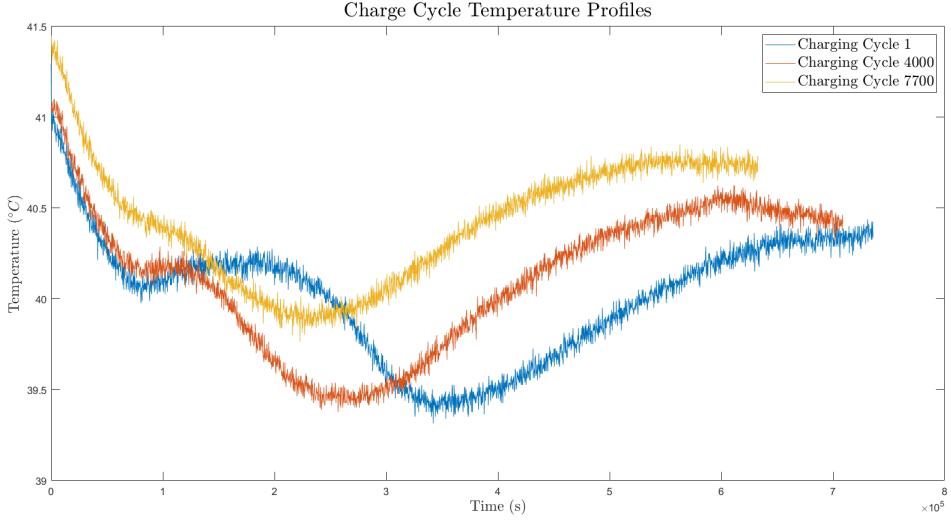


Figure 4.2: The charging temperature profiles of 3 cycles (the 1st, 4000th, 7700th). This shows how the maximum charging temperature changes with cycles but also the entire charging profile.

The calculation of the posterior of Q_t would be defined by

$$P(Q_t|T_t) = \frac{P(Q_t, T_t)}{P(T_t)} \quad (4.2)$$

where

$$P(T_t) = P(T_t|Q_{t-1}) \quad (4.3)$$

and the prior of $P(Q_t)$ would be defined by propagating particles and performing KDE on them.

Then, similarly, for T_{t+1} , first, particles would be sampled from its previous distribution $P(T_t|Q_{t-1})$ and propagated stochastically with the particle variance to form $P(T_{t+1})$ and using the conditional formula as seen in equation 4.2, the posterior is calculated as

$$P(T_{t+1}|Q_t) = \frac{P(T_{t+1}, Q_t)}{P(Q_t)} \quad (4.4)$$

where $P(Q_t)$ is defined in equation 4.2.

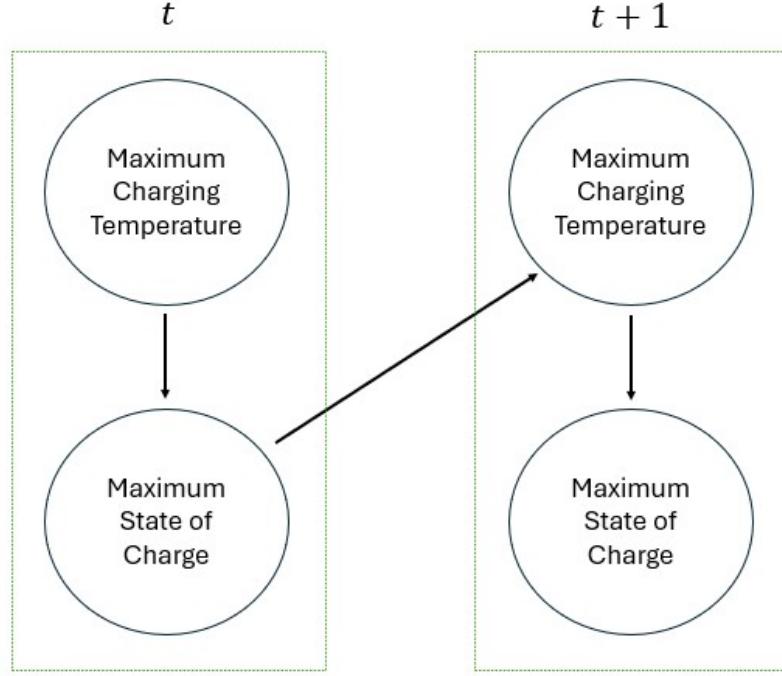


Figure 4.3: The basic DBN for battery degradation prediction featuring two variables

For a timeslice that contains observation data, the posterior from a non-observation timeslice may be referred to as the pseudo-posterior as in algorithm ???. The formulas for the posteriors hold but now need a correction due to the observation data. The new maximum charge posterior for calculation is $P(Q_t|T_t, O_{Q,t})$ given by Bayes' theorem as follows

$$P(Q_t|T_t, O_{Q,t}) \propto P(O_{Q,t}|Q_t) \cdot P(Q_t|T_t). \quad (4.5)$$

where $P(O_{Q,t}|Q_t)$ is the likelihood of an observation of maximum state of charge given by equation 4.1 and where $P(Q_t|T_t)$ is the prior or, as defined in algorithm ???, the pseudo-posterior.

Similarly, the posterior of the maximum charging temperature, $P(T_{t+1}|Q_t, O_{T,t+1})$ can be calculated also using Bayes' theorem as follows

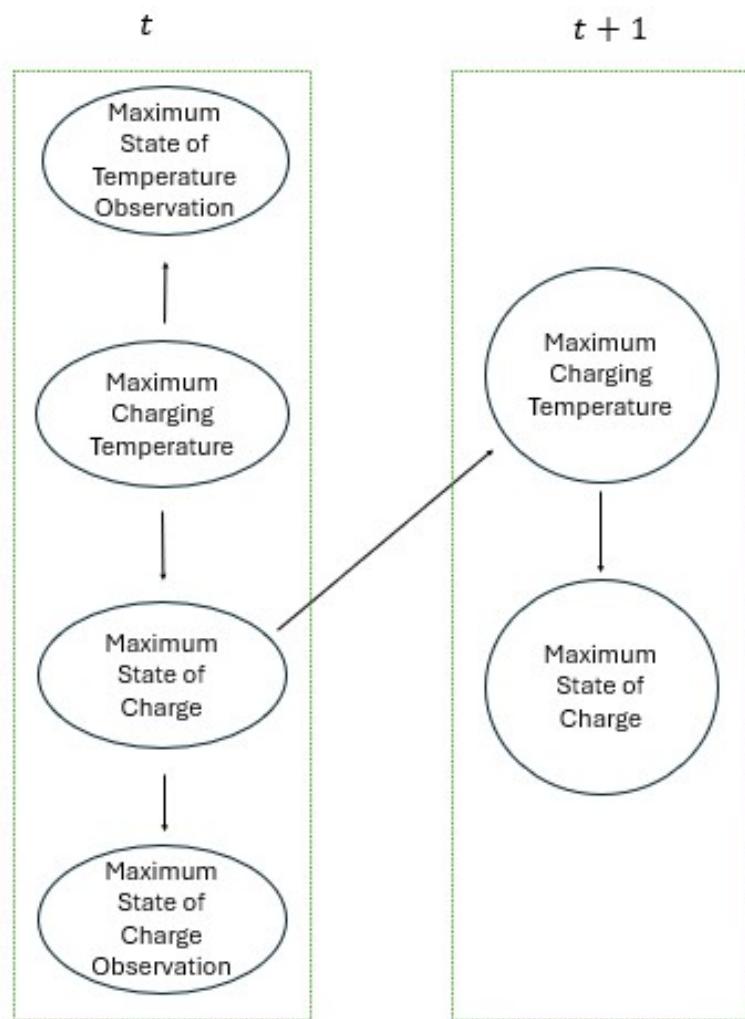


Figure 4.4: Modified DBN from 4.3 to include the incorporation of intermittent observation data present at timeslice t .

$$P(T_{t+1}|Q_t, O_{T,t+1}) \propto P(O_{T,t+1}|T_{t+1}) \cdot P(T_{t+1}|Q_t). \quad (4.6)$$

Within the PF algorithm used in this thesis, the posterior given the observations would be calculated as weights and subject to normalization as detailed in algorithm ??.

Furthermore, due to the presence of two hidden variables, this particular network would not require the adaptive grid KDE due to the relatively low computational complexity of calculating a joint distribution of two variables as opposed to that of four and beyond. However, the “leave-one-out” cross validation method for bandwidth selection can be incorporated into this method.

4.4 Results

The following section details the results of the basic network with fifty particles. The simulation used the cross-validation method for bandwidth selection in the KDE and did not use the adaptive gridding due to the relatively low computational complexity. The results feature a short term RUL indicator where the particles were propagated in steps of twenty-five cycles and received a measurement update every 100 cycles. The data featured cycles up to 7700 cycles and so the short term RUL indicator ceases here. The results also feature a long term RUL indicator where the particles were propagated in steps of 100 cycles i.e. receiving a measurement update at every step. The particles were then propagated forward to 10000 cycles and the mean prediction and the prediction intervals are featured. Finally, the evolution of the belief is featured in ridge plots showing how the belief of future charges change as time progresses showing the learning of the algorithm.

4.4.1 Short Term RUL Indicator

As said, these graphs feature the particles being propagated at 25 cycle intervals with correction at every 100 cycles. The graph of the maximum state of charge with these features is seen in figure 4.5. The expansion of the prediction intervals shows the rapid growth in uncertainty which occurs with stochastic propagation of particles over time. The somewhat uniformity of the prediction intervals from, approximately, the 6000th cycle to the 7500th cycle may be indicative of the model converging, however, due to the non-uniform movement of the informative variable, the maximum charging cycle temperature as seen in figure 4.6, this is likely more as a result of the Markov property whereby it is learning from its prior movements.

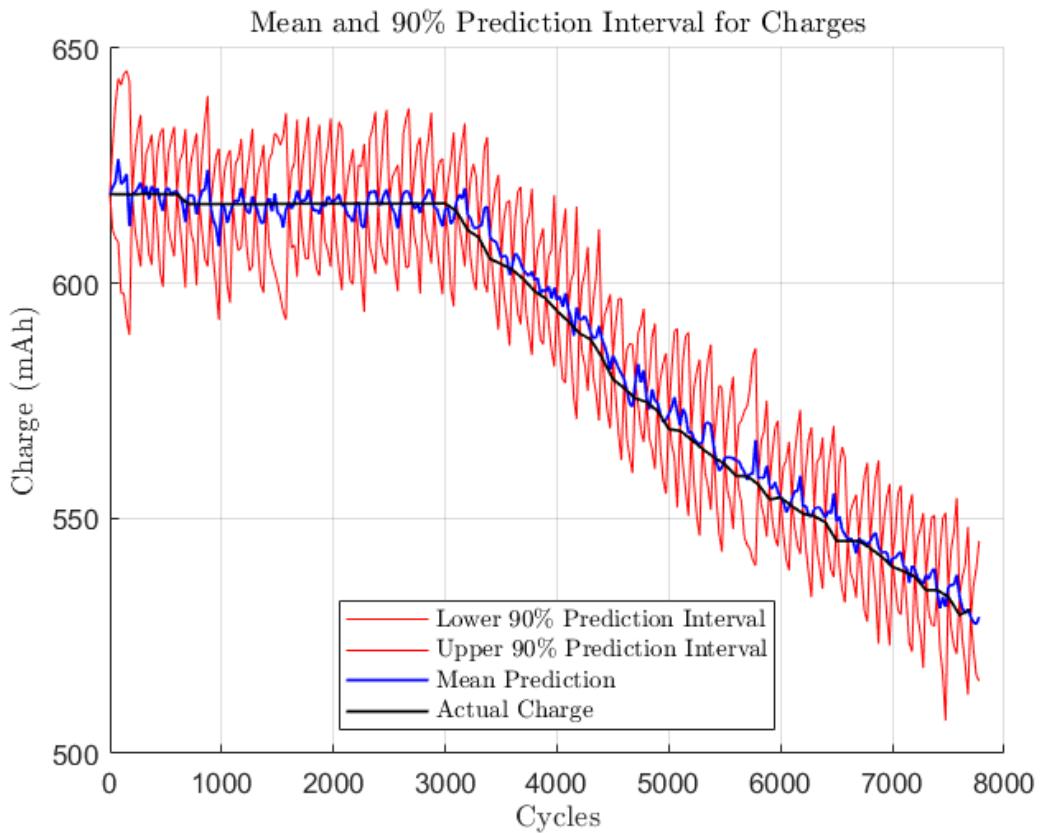


Figure 4.5: Plot showing the actual maximum charge, the mean prediction and the upper and lower 90% prediction intervals.

Figure 4.6, shows the propagation and correction of the informative parent variable. Due to the non-uniform nature of this variable, with frequent up and down fluctuations, this as a parent variable does not help our query variable converge. The fluctuations may be due to poor measurement equipment as part of the experimental apparatus which could lead to the increase of particle noise in the likelihood measurements. This in turn could have led to a converged predictive model.

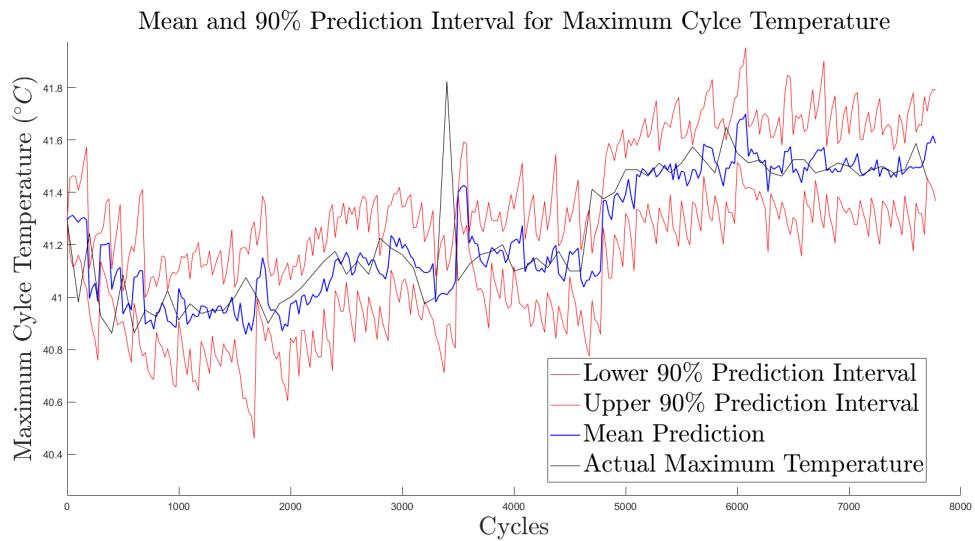


Figure 4.6: Plot showing the actual maximum charging temperature, the mean prediction, and the upper and lower 90% prediction intervals.

4.4.2 Evolution of Belief

When analysing figures 4.5 and 4.6, the lack of distribution can make it difficult to understand exactly how the beliefs of the predictive model evolve over time. Seeing the predictive intervals can help to visualize the uncertainty in the predictive model but to see the specific characteristics of a variables' distribution requires the ridge plots as seen in figures 4.7 and 4.8.

Figure 4.7 shows the multimodal, characteristic distribution of the maximum state of charge of the battery. The increased height in the graph on the right, with cycles

7250 to 7750, shows the increased confidence of the model with prediction intervals narrowing slightly as also seen in figure 4.5. That said, the lack of uniformity at any stage of the battery life shows a predictive model that has not converged. Again, this is likely due to the large fluctuations felt by this query node from its parent node, the maximum cycle temperature.

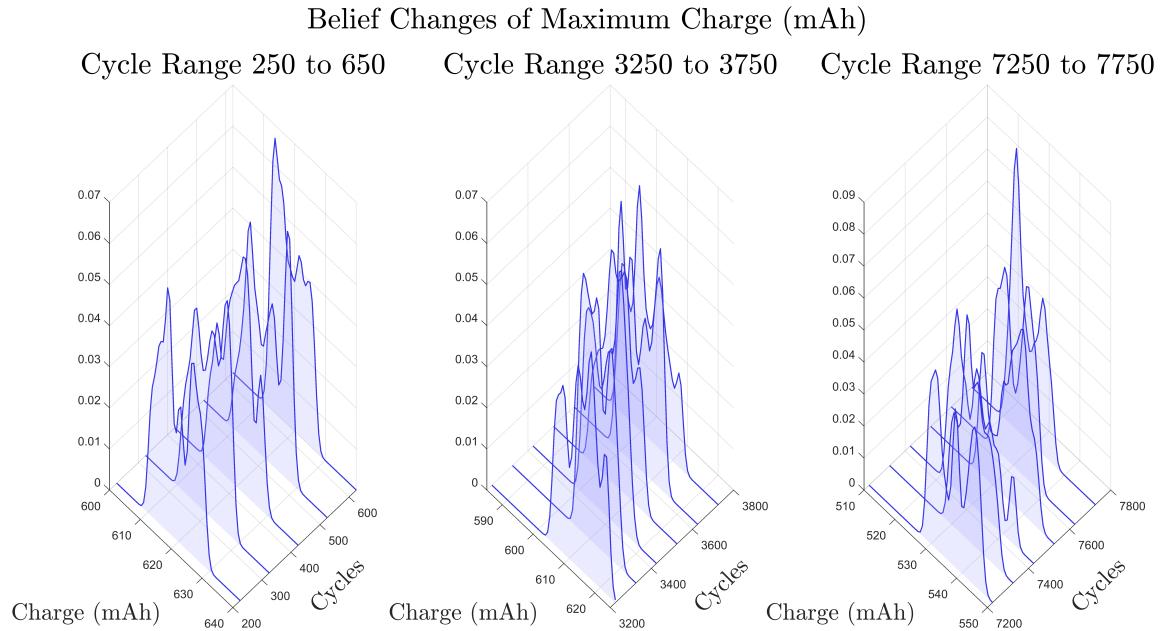


Figure 4.7: Ridge plot showing the evolution of the maximum charge distribution across the many charging cycles. Each of these ridges were 25 cycles after a measurement update and show evolution of the distributions.

Figure 4.8 shows the evolution of the maximum charging cycle temperature from no real pattern in the distribution to a stronger more consistent distribution in the latter cycles when comparing the three graphs. This shows convergence of the predictive model for this variable and may be as a result of the more uniform nature of the maximum state of charge of the battery which is the parent node for this variable.

An important point to make is that despite the lack of convergence of the results on a uniform distribution and uncertainty growth rate, a characteristic, multi-modal distribution is seen which is favourable for the predictive model showing its ability to

evolve upon the availability of new data.

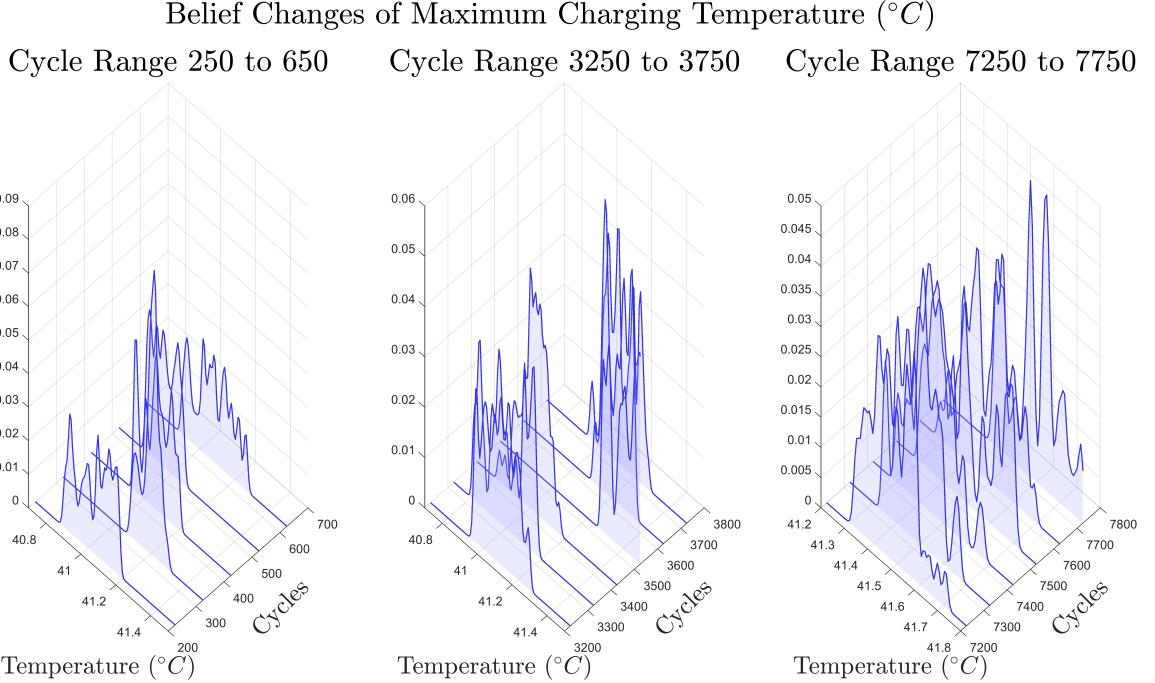


Figure 4.8: Similar to 4.7, the evolution of the maximum charging cycle temperature can be seen with each ridge showing the distribution 25 cycles after a measurement update.

4.4.3 Long Term RUL Indicator

The main goal of the long term RUL indicator graphs is to visualize the propagation of uncertainty after final measurement updates. From cycle 7700 to cycle 10000, the unsupervised propagation of particles is achieved through the continuous sampling of the distribution. This causes the distribution's prediction intervals to spread as seen in both figures 4.9 and 4.10.

Figure 4.9 shows the expansion of the upper bound far more than the lower bound. This shows the incorrect learning of the network as one knows that a battery can not reverse its degradation. The mean prediction also shows a reversal of the degradation process. As previously stated, the fluctuations in maximum temperature was likely

the cause of the lack of convergence of the charge predictive model, and it's likely that it caused the reversed prediction for the degradation of the battery. This is due to the slow fall in maximum temperature from cycle 5000 onward. Since the general relationship was that an increase in temperature resulted in battery degradation, the model then sees a fall in maximum temperature and sees it as an increase in maximum state of charge going forward toward cycle 10000.

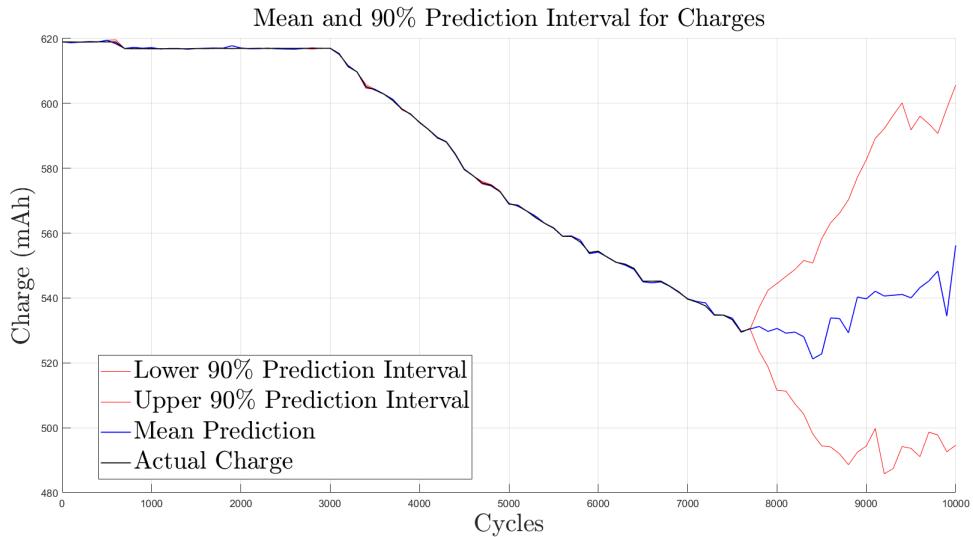


Figure 4.9: Plot showing the actual maximum charge, the mean prediction and the upper and lower 90% prediction intervals. These are then propagated beyond available data to show the model's prediction and the propagation of uncertainty.

As said previously, the general relationship was that an increase in temperature resulted in battery degradation, and looking at figure 4.10, the mean prediction shows a slow fall continuing up to cycle 10000, with the lower prediction bound stretching more than the upper. It may also be as a result of the final tick down seen but the since the fall is slow, it's more likely as a result of the slow fall seen from cycle 5000 onward.

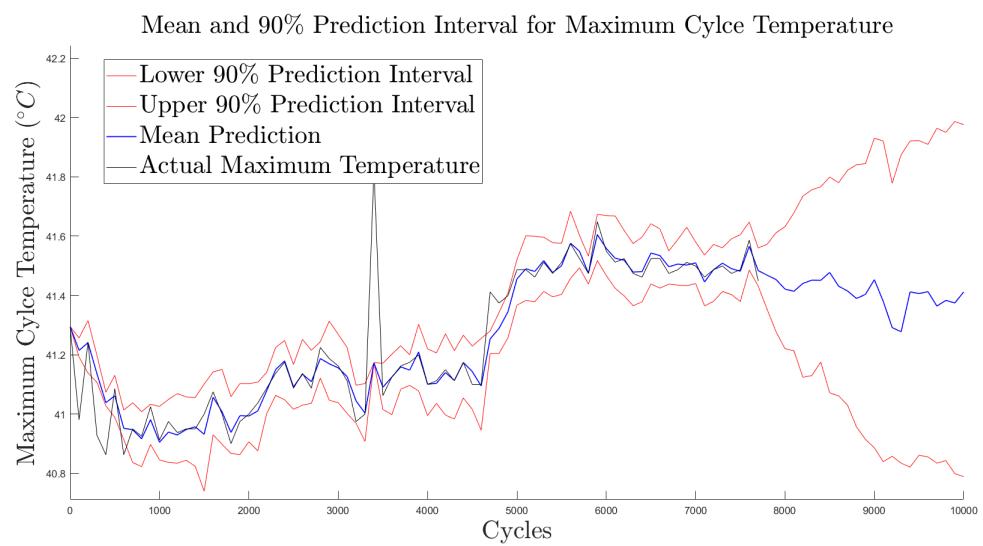


Figure 4.10: Plot showing the actual maximum charging temperature, the mean prediction and the upper and lower 90% prediction intervals. These are then propagated beyond available data to show the model's prediction and the propagation of uncertainty.

Chapter 5

Complex Network for Battery Degradation

The same test data for this network was used as for the basic network.

5.1 Proposed Network

The proposed network adds to the complexity of the previous network by incorporating two more parent nodes. This means that the new parent nodes now consist of maximum charging temperature, maximum charging voltage, and charging time, and maximum state of charge as the child, query node as seen in figure 5.1. The choice to have these as parent nodes speaks to their ability to act as battery degradation indicators.

As discussed in chapter 4, the maximum temperature has a complex relationship with battery degradation as seen in the third graph of figure 5.2 but can still be incorporated into a life cycle model.

The two added variables to this network, maximum charging voltage and charging time act as stronger indicators when analysing the first two graphs in figure 5.2. For both the changes to the charge time and the maximum charging voltage are as a

result of increased internal resistance caused by the degradation of the electrodes and the loss of lithium inventory as seen in figure 5.2. Despite the increased maximum voltage, this is only achieved as a result of reduced battery capacity which allows the voltage to reach higher thresholds. The increased battery charging time is caused by the increased internal resistance caused by a battery's degradation.

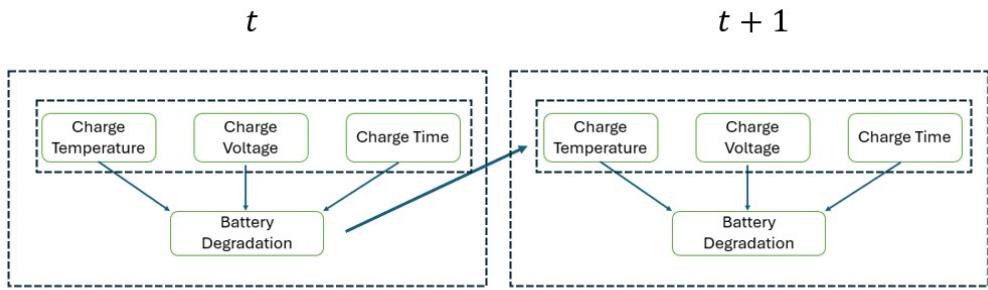


Figure 5.1: Complex network with two added variables.

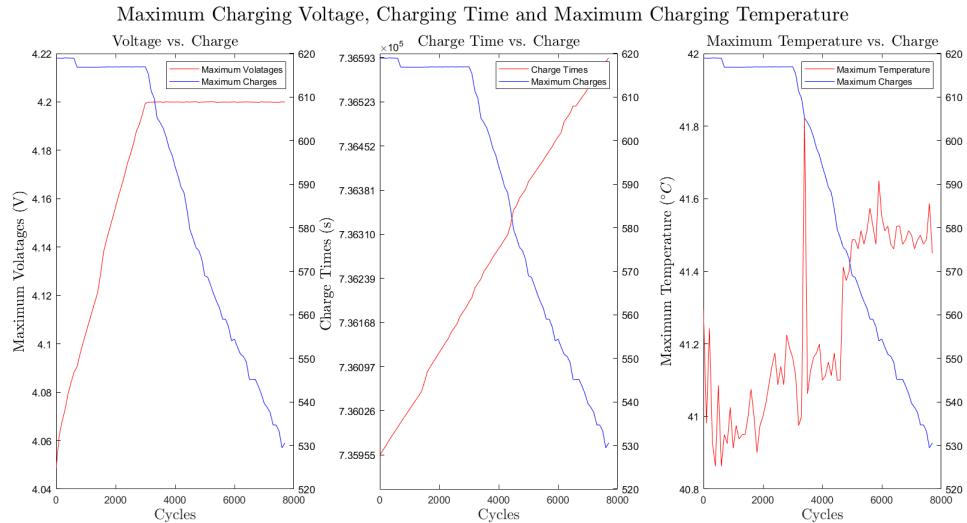


Figure 5.2: (From Left to Right) Maximum charging voltage and the maximum charge, charging time and the maximum charge, maximum charging temperature and the maximum charge. All plotted against the corresponding charge cycle.

5.2 Proposed Algorithm II

The algorithm for the more complex network features extensions to the probability equations and introduce the need for a more efficient method of evaluating KDEs due to the significantly larger computational efforts required for the increase in dimensions. This paves the way for the application of the adaptive grid KDE as described in section 3.2.3. The changes to the psuedo-posterior and posterior feature simple extensions in the case of the maximum charge node and the pseudo-posterior and posterior for the maximum charging voltage and the charging time are very similar to that of the maximum charging temperature in the basic network.

In the case of timeslices that don't feature observation nodes, the pseudo-posterior need only be calculated. At timeslice t , let Q_t denote maximum charge, T_t denote maximum charging temperature, V_t denote maximum charging voltage, and τ_t denote charging time for a cycle. The pseudo-posterior of Q_t may be calculated by the conditional probability formula which places the joint distribution over the marginal distribution as follows

$$P(Q_t|T_t, V_t, \tau_t) = \frac{P(Q_t, T_t, V_t, \tau_t)}{P(T_t, V_t, \tau_t)}. \quad (5.1)$$

where the parent nodes would have been sampled from previous distributions obtained like the maximum charging temperature in the basic network. Considering the previous timeslice, $t - 1$, then the following pseudo-posteriors would hold for the parent nodes that form the joint and marginal distributions in equation 5.1:

$$P(T_t) = P(T_t|Q_{t-1}) = \frac{P(T_t, Q_{t-1})}{P(Q_{t-1})}, \quad (5.2)$$

$$P(V_t) = P(V_t|Q_{t-1}) = \frac{P(V_t, Q_{t-1})}{P(Q_{t-1})}, \quad (5.3)$$

and

$$P(\tau_t) = P(\tau_t|Q_{t-1}) = \frac{P(\tau_t, Q_{t-1})}{P(Q_{t-1})}. \quad (5.4)$$

Considering a timeslice where an observation is available, the pseudo-posterior is updated to consider the likelihood of the observation given the model output to form the posterior, or the probability of the pseudo-posterior given the observation. Since the observation nodes only inform their parent nodes directly and no other, the likelihood PDF can be calculated on a univariate case as opposed to a multivariate whereby a node may be informed by multiple observation nodes, as seen in factorial HMMs.

In this case, the nodes are updated just as they are in the basic network and then subject to normalisation as described in algorithm ???. These updates of the pseudo posterior look as follows:

$$P(Q_t|T_t, V_t, \tau_t, O_{Q,t}) \propto P(O_{Q,t}|Q_t) \cdot P(Q_t|T_t, V_t, \tau_t), \quad (5.5)$$

and for the parent nodes:

$$P(T_t|Q_{t-1}, O_{T,t}) \propto P(O_{T,t}|T_t) \cdot P(T_t|Q_{t-1}), \quad (5.6)$$

$$P(V_t|Q_{t-1}, O_{V,t}) \propto P(O_{V,t}|V_t) \cdot P(V_t|Q_{t-1}), \quad (5.7)$$

and

$$P(\tau_t|Q_{t-1}, O_{\tau,t}) \propto P(O_{\tau,t}|\tau_t) \cdot P(\tau_t|Q_{t-1}). \quad (5.8)$$

5.3 Results

5.3.1 Short Term RUL

Like section 4.4.1, this section of results shows the plot of each variable at 25 cycle intervals with prediction means and prediction intervals as well as the actual value. Figure 5.3 shows this for the maximum state of charge. The confidence intervals can be seen to be far more uniform than the two variable network.

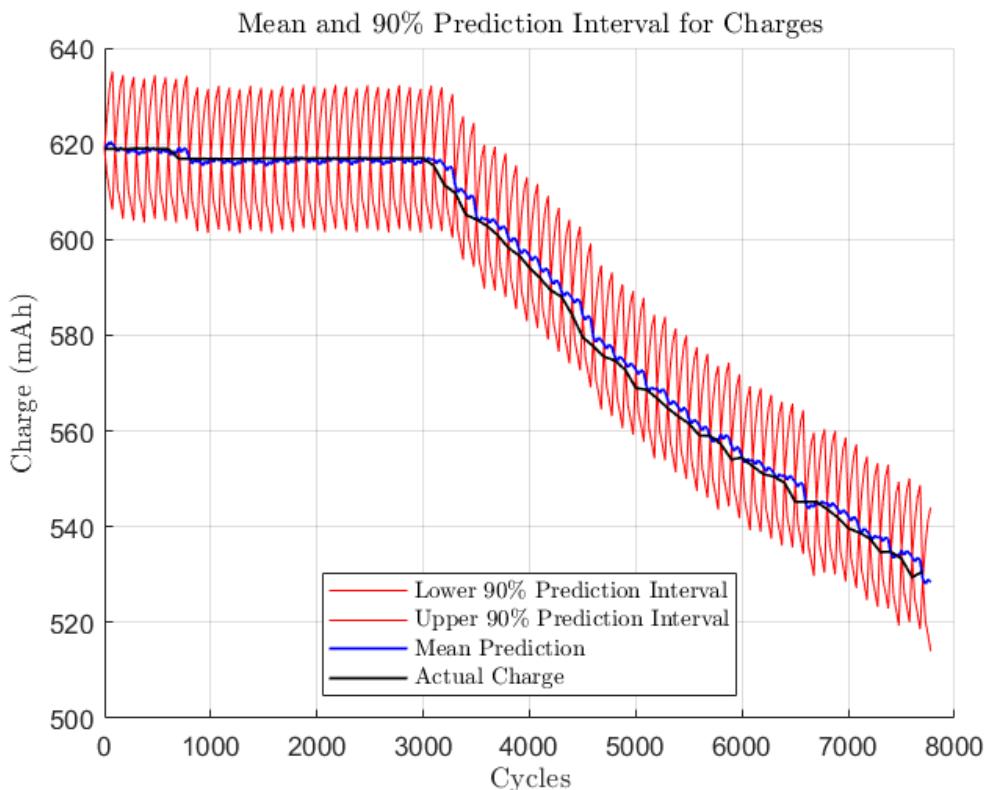


Figure 5.3: Plot showing the actual maximum charge, the mean prediction and the upper and lower 90% prediction intervals from the complex network with four variables.

Similarly, the maximum temperature plot, in figure 5.3, is more uniform than its two variable counterpart.

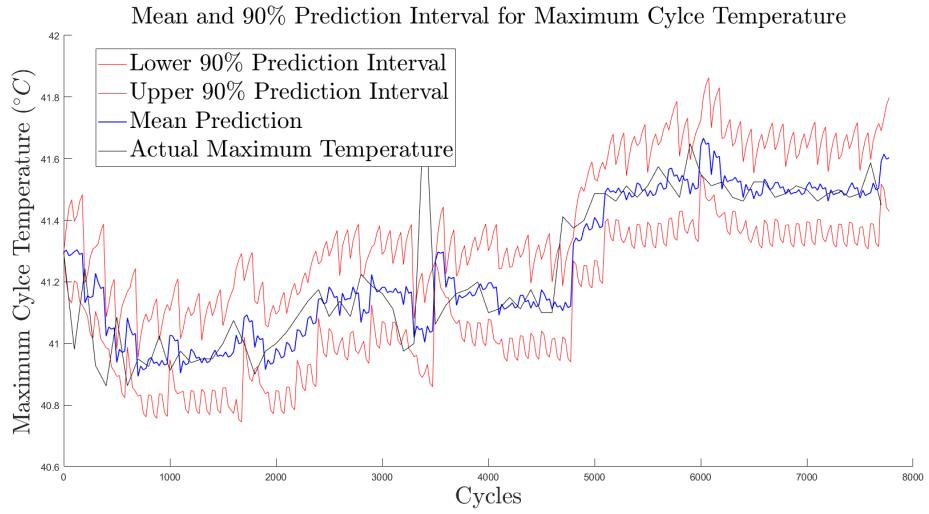


Figure 5.4: Plot showing the actual maximum charging temperature, the mean prediction and the upper and lower 90% prediction intervals from the complex network with four variables.

Figures 5.5 and 5.6 show the maximum voltage plot and charging time, respectively, with their mean predictions and prediction intervals. The uniformity of the actual values indicate how they were able to stabilise the predictive model and allow for more uniform uncertainty quantifications.

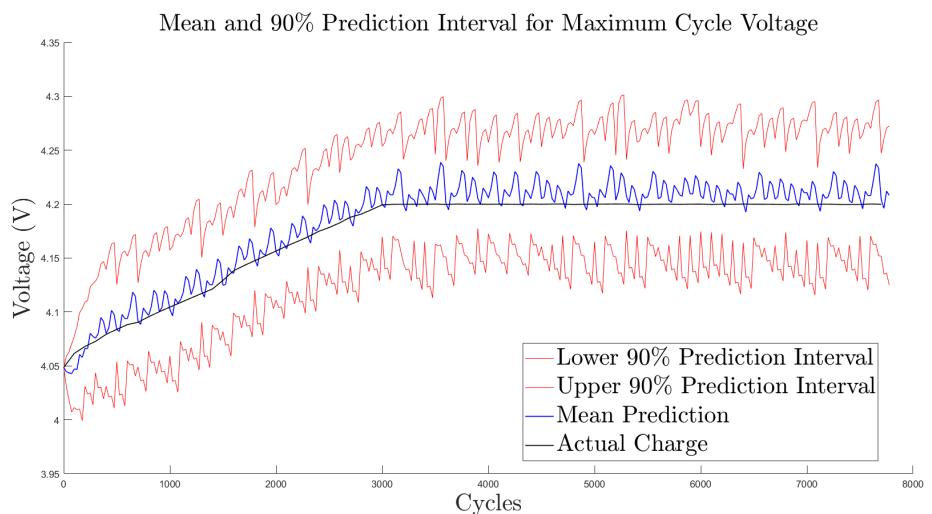


Figure 5.5: Plot showing the actual maximum voltage, the mean prediction and the upper and lower 90% prediction intervals from the complex network with four variables.

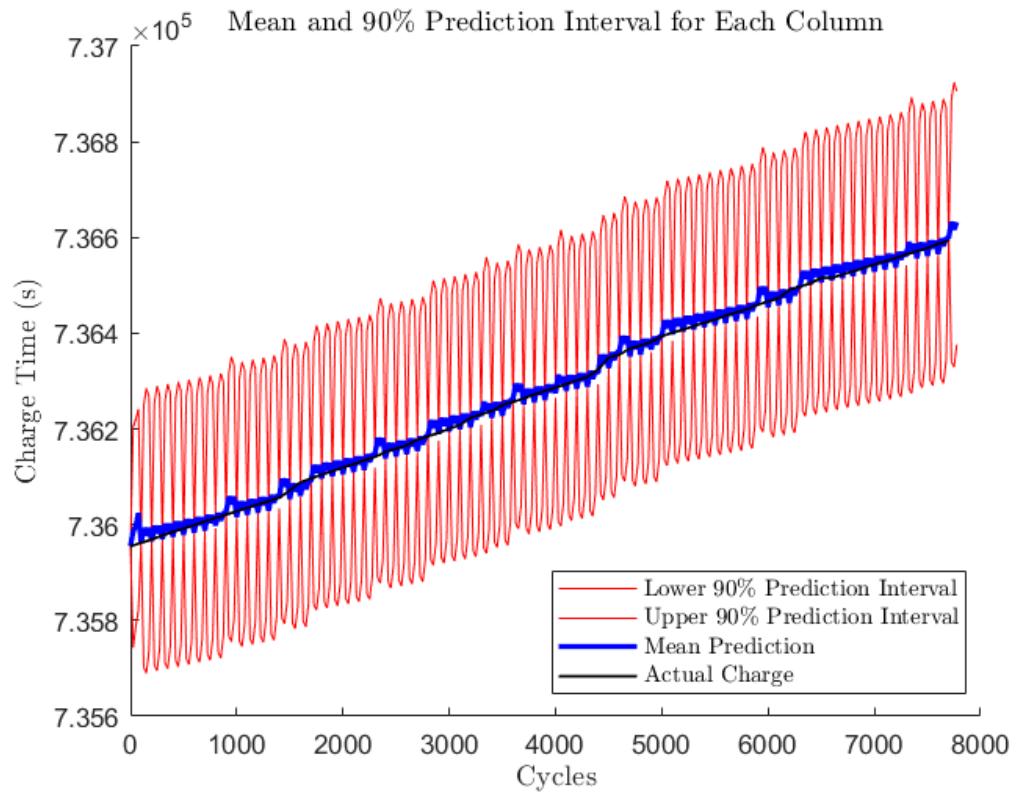


Figure 5.6: Plot showing the actual charging time, the mean prediction and the upper and lower 90% prediction intervals from the complex network with four variables.

5.3.2 Evolution of Beliefs

The evolution of beliefs helps visualize the uncertainty quantification and this networks convergence as the one sees the uniformity hold in these distributions in figures 5.7, 5.9 and 5.10. Of course, the exception to this is 5.8 which still suffers from a varying belief distribution and jagged prediction intervals due to its complex relationship with the other variables.

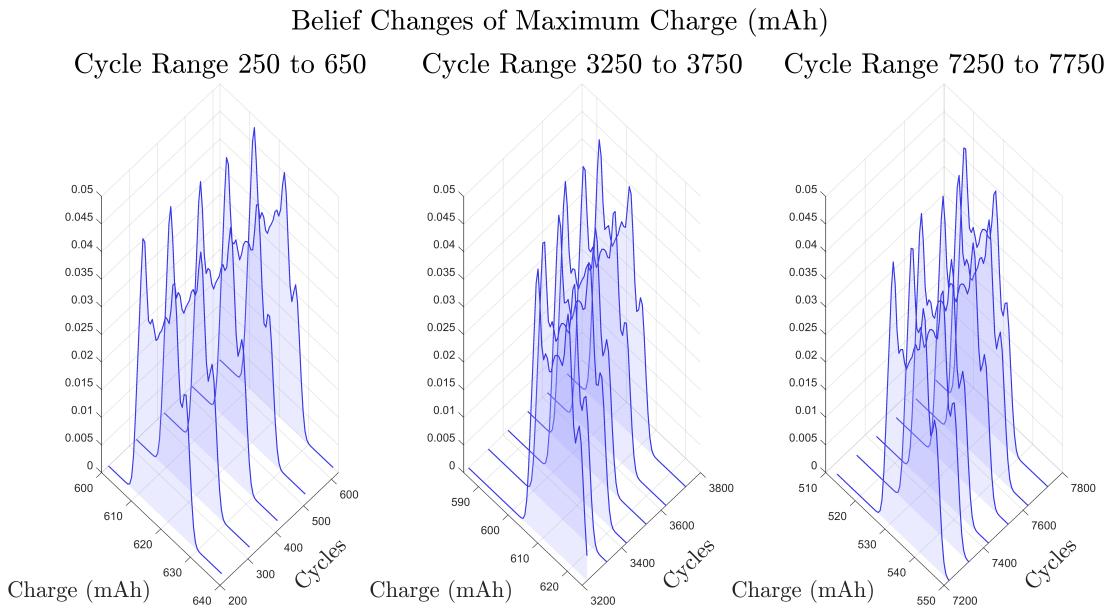


Figure 5.7: The evolution of the maximum charge from observation to observation and across the full span of the battery life showing the evolution of the belief.

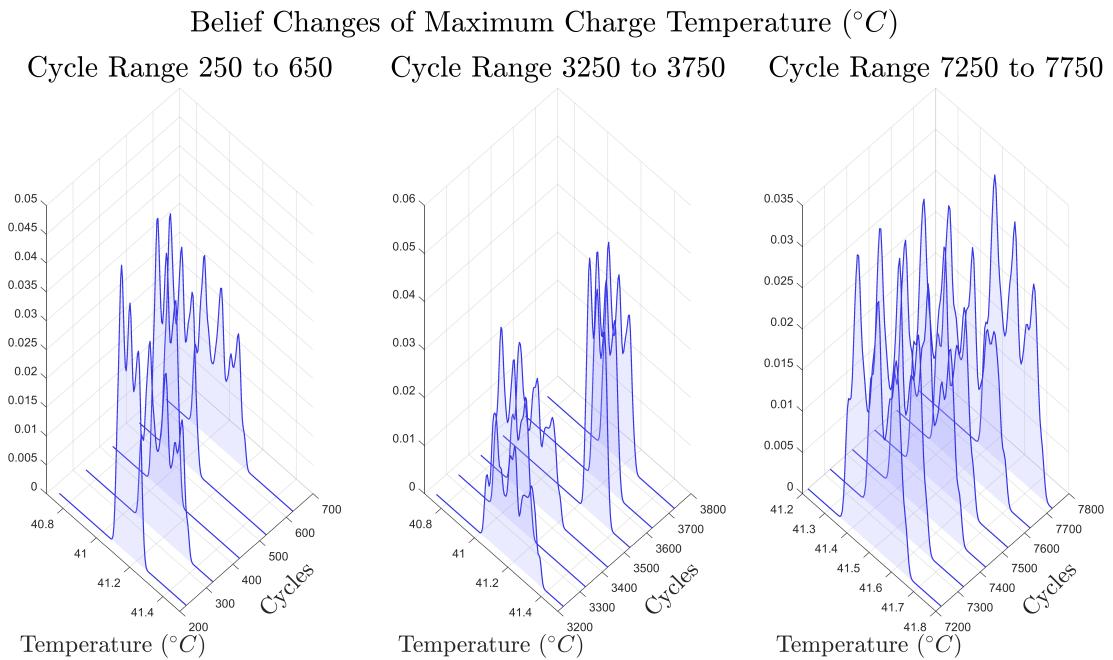


Figure 5.8: Similar to figure 5.7, the evolution of the maximum charging temperature from observation to observation and across the full span of the battery life showing the evolution of the belief.

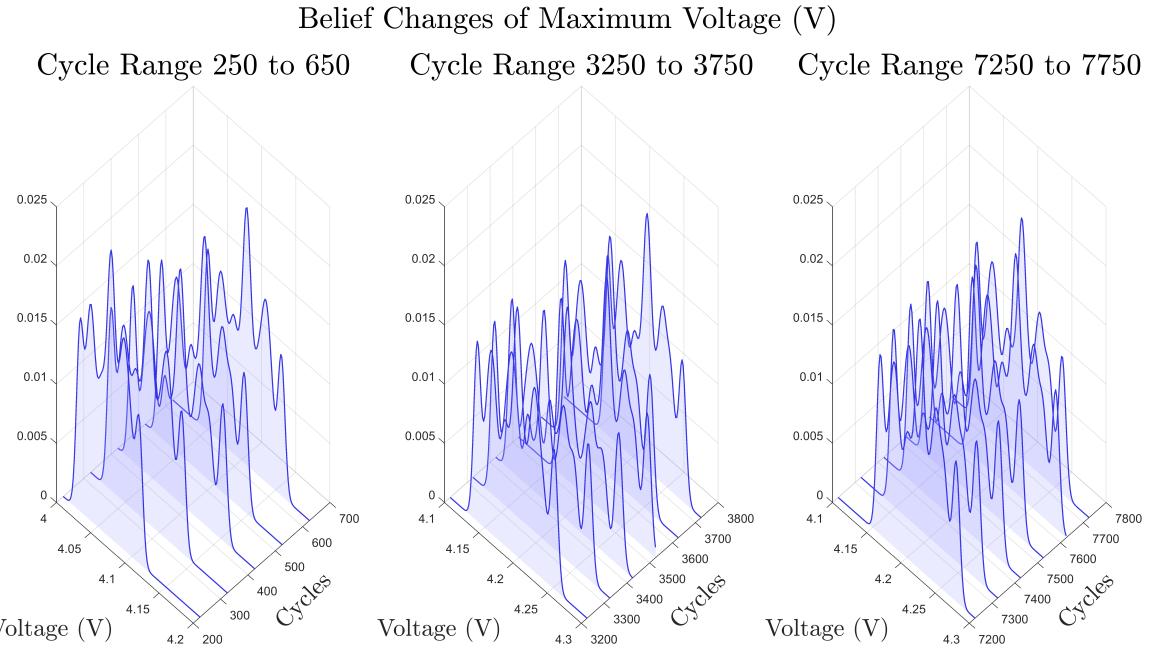


Figure 5.9: The evolution of the maximum charging voltage belief from observation to observation and across the full span of the battery life.

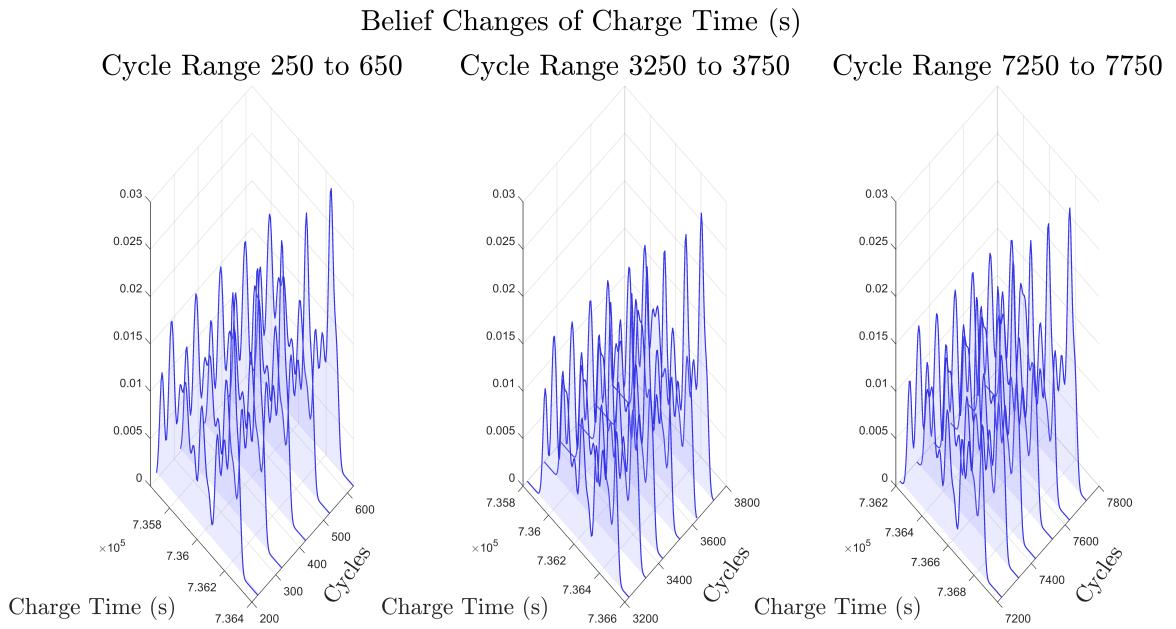


Figure 5.10: The evolution of the charging time belief from observation to observation and across the full span of the battery life.

5.3.3 Long Term RUL

When considering the long term predictions of the variables up to 10000 cycles, figures 5.11, 5.12, 5.13 and 5.14, one sees key characteristics in the predictive properties of the model from cycle 7700 onward.

The long term charge indicator is more promising than that of its counterpart in the basic network with two variables due to the upper bound sloping downward. This captures the model's recognition of the battery degradation as it can not reverse its degradation. The prediction mean also closely tracks the true degradation trajectory, demonstrating strong predictive accuracy and reliable long-term behavior.

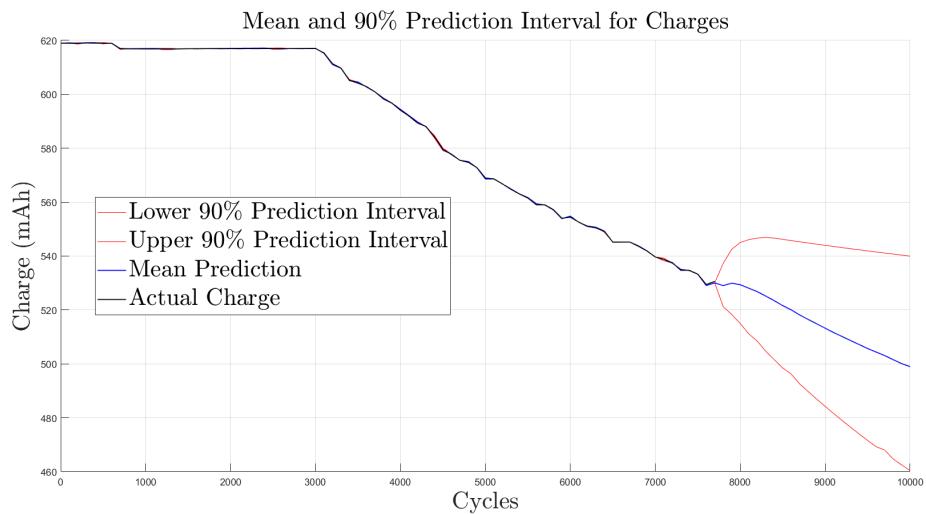


Figure 5.11: Plot showing the actual maximum charge, the mean prediction and the upper and lower 90% prediction intervals. These are then propagated beyond available data to show the model's prediction and the propagation of uncertainty.

Similarly, in figure 5.12, differences can be seen with the temperature prediction and the prediction interval, with a steady increase in maximum charging temperature forecasted, this agrees with the rough relationship between maximum charge and temperature having an inverse relationship. The width of the prediction intervals and their steady uniformity indicates a more fixed uncertainty quantification.

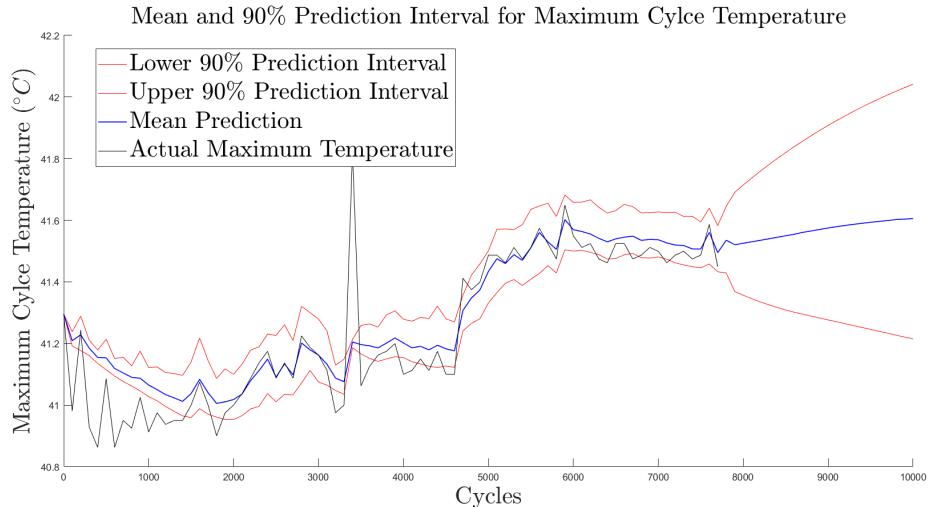


Figure 5.12: Plot showing the actual maximum charging temperature, the mean prediction and the upper and lower 90% prediction intervals. These are then propagated beyond available data to show the model's prediction and the propagation of uncertainty.

Figure 5.13 shows the prediction of a fall in internal voltage as seen by both the prediction mean and the prediction intervals. This isn't impossible like the reverse degradation as a batteries often increase their voltage threshold while degrading but can have a decrease later in their life.

Lastly, the charge time, in figure 5.14, predicts a sharper increase in charge time potentially caused by the fall in voltage or the steady increase in temperature predicted.

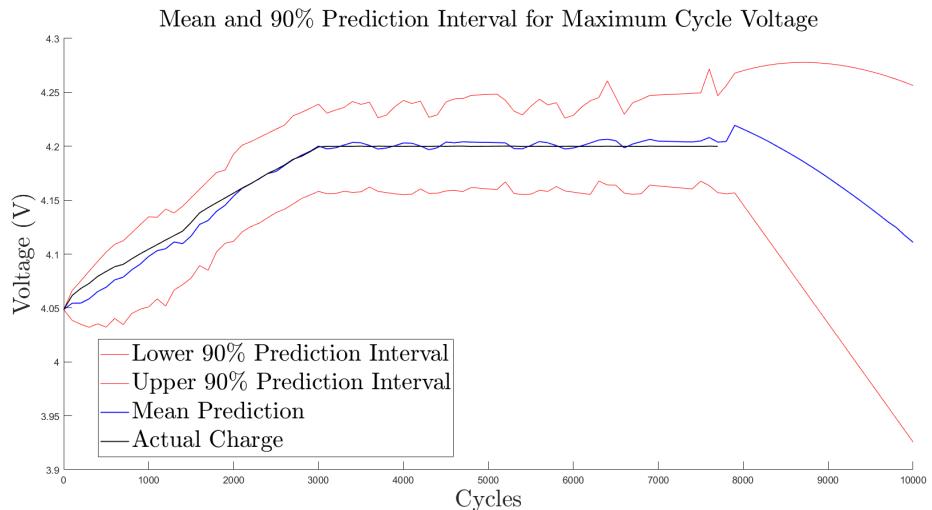


Figure 5.13: Plot showing the actual maximum charging voltage, the mean prediction and the upper and lower 90% prediction intervals. These are then propagated beyond available data to show the model's prediction and the propagation of uncertainty.

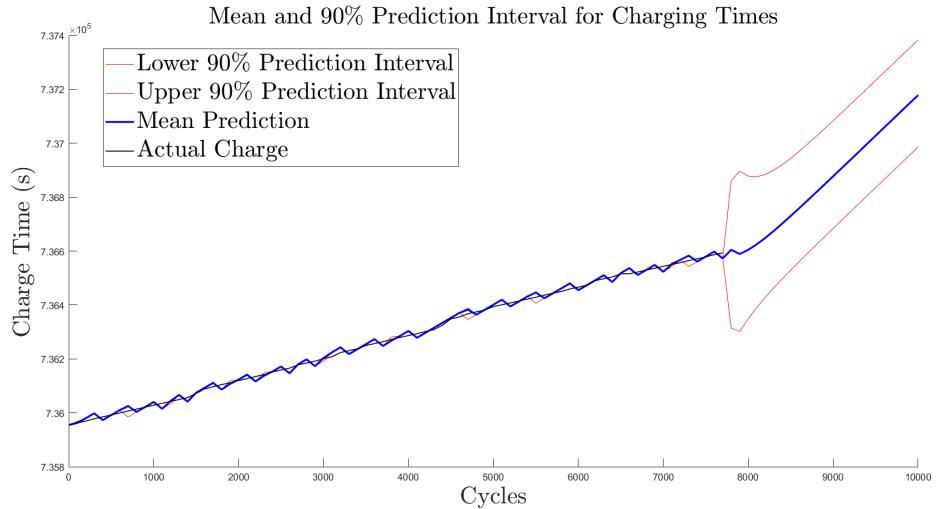


Figure 5.14: Plot showing the actual charging time, the mean prediction and the upper and lower 90% prediction intervals. These are then propagated beyond available data to show the model's prediction and the propagation of uncertainty.

Chapter 6

Discussion

6.1 Validity of Results

When using a particle filter as a method of inference, a serious problem that can arise is sample degeneracy, as mentioned in section 2.2.3, whereby due to a lack of variance with a variable, the same particle will be repeatedly sampled through the weighted sampling mechanism due to significantly higher probability being associated with a particle. Due to the KDE transition model, a higher amount of control may be placed over the smoothness of the distribution by using appropriate bandwidth selectors as detailed previously in sections 2.3.1 and 3.2.1. The cross validation method, using ISE in this case, can help with mitigating sample degeneracy by reducing over confidence in high density regions. Furthermore, in the case of the complex network, the use of the adaptive grid helps for increased diversity in high density regions with increased resolution being able to capture the multi-modality of the region in more detail.

In order to test this for this particular case, the ESS can be used, first proposed by (Kong 1992), whereby the optimal value is the sample size, or in this case, the number of particles, N , and total degeneracy is 1. The ESS is calculated by

$$ESS = \frac{1}{\sum_{i=1}^N w_i^2}, \quad (6.1)$$

where w is the normalized weight associated with the i^{th} particle.

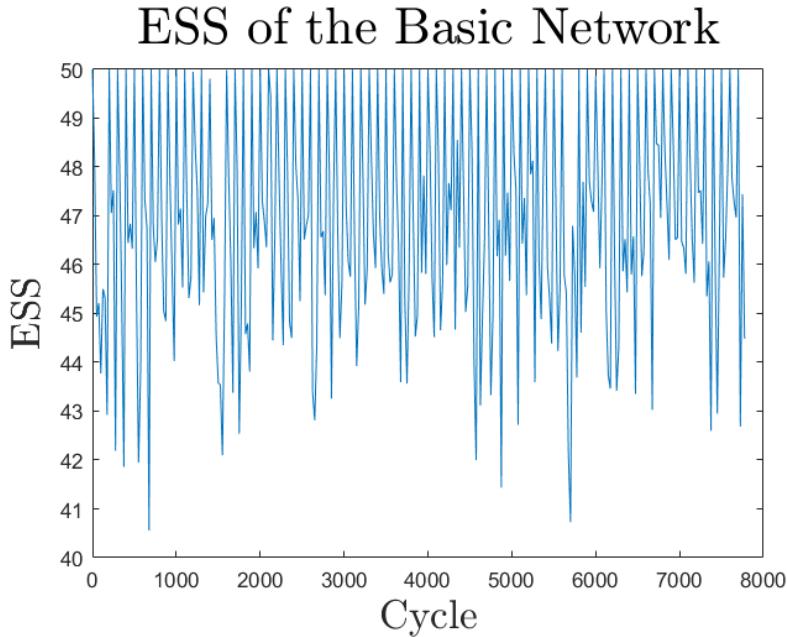


Figure 6.1: Plot showing the ESS for the basic network across all timeslices (0 to 7700)

When applying this to each timeslice of each network, the resulting plots in figures 6.1 and 6.2 for both the basic and complex network, respectively. The ESS in both frequently hits the value for N (50 in the case of the basic network and 40 in the case of the complex network) upon measurement updates. This is highly degenerate despite the high value but the weights represent such a narrow region of values, all weights are almost equal. Twenty-five cycles after a measurement update, the ESS falls due to propagation of particles which occupy a lower volume than later on and so a high density PDF region forms. The further increase there after is as a result of dimensionality increasing the data sparsity but also spreading the density region more evenly as prediction intervals tend to spread wider at these points.

The ESS test concludes that particle degeneracy was mitigated successfully owing to a combination of a sufficient amount of particles used for each network, and a successful selection of bandwidth by the cross validation method.

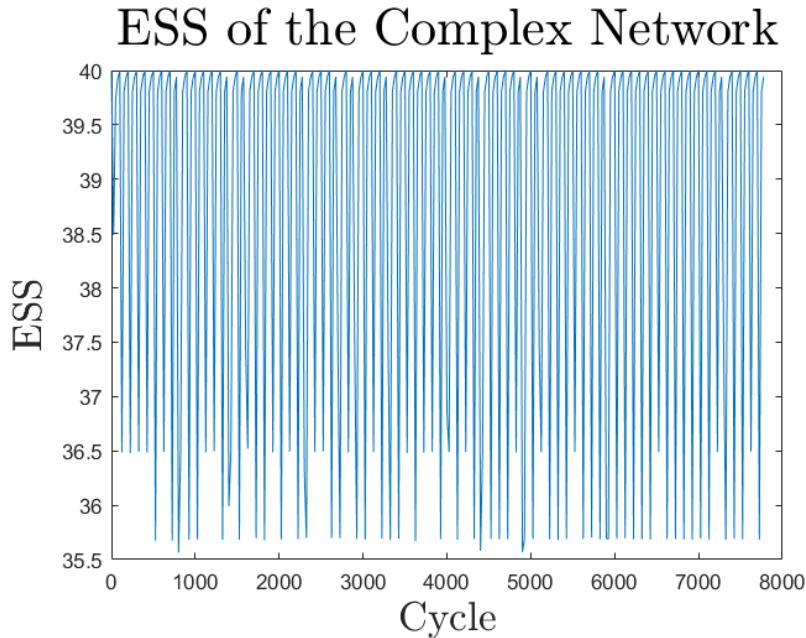


Figure 6.2: Plot showing the ESS for the complex network across all timeslices (0 to 7700)

Moreover, within the results contained in chapters 4 and 5, various graphical signs show sufficient sample diversity and convergence of the predictive model. The wide and steady prediction intervals existent in variables for both the basic and complex network show both convergence and particle trajectories to be diverse. When looking at the evolution of belief plots in sections 4.4.2 and 5.3.2, the multi-modality of the plots also show the diversity of the particles present. The evolution plots also show the distributions 25 cycles after a measurement update which is when the particles are least diverse and yet these distributions still show sufficient diversity.

6.2 Comparison of Networks

When comparing the results of both networks, key differences in characteristics emerge in all three types of results and between the corresponding variables in both networks. In general, the results are more favourable for the complex network. The indications that lead to this conclusion are detailed in the following sections.

6.2.1 Short Term RUL Indicators

Contained in sections 4.4.1 and 5.3.1 the short term results of both networks with 25 cycle intervals and measurement updates every 100 cycles.

The main differences seen between these networks is the charge prediction intervals which are far more stable and uniform in the complex network rather than the more jagged, unpredictable prediction intervals of the basic network. This shows convergence and more definite uncertainty quantification signaling the convergence of the complex network, more so than the basic network.

When comparing the same features in the maximum temperature plot, the maximum temperature plot for the complex network is slightly more uniform than its basic network counterpart. The reason for the greater stability difference in the charge plot is due to the DBN structure, where temperature is only ever reliant on the charge particles. Whereas, the charge particles are now informed by two more variables. This added stability of the charge is able to stabilise the temperature somewhat but not as much as how stable the charge becomes.

When analysing these two added variables in the complex network, namely, the maximum voltage and the charging time, it's clear the stability is owed to the predictive nature of these variables as opposed to the maximum charging temperature which is far more complex and unpredictable.

6.2.2 Differences in Evolution of Beliefs

Building on the previous section, the more consistency of prediction intervals can also be seen in the differences in the belief plots seen in sections 4.4.2 and 5.3.2. In the case of the basic network, for both variables, the evolution of the beliefs are far more extreme and have severe changes from one plot to another. Again, this shows a lack of certainty in the quantification of uncertainty around these variables and a lack of convergence of the basic network relative to the complex network.

In contrast, the belief plots in section 5.3.2 are far more consistent and only vary slightly through timeslices. This shows the predictive models convergence only needing to evolve slightly from timeslice to timeslice and gives a far greater confidence in its uncertainty quantification.

6.2.3 Long Term RUL Indicators

The key differences in the long term indicators in sections 4.4.3 and 5.3.3 lay in the predictions from cycle 7700 onward where the learned distribution of the variables propagate particles forward.

The general differences between both networks results across all variables are the jagged prediction intervals in the basic network versus the smooth prediction intervals of the complex network. Generally, the smooth prediction intervals indicate a confident and coherent model that has converged. It is also reasonable to assume the uncertainty is correctly quantified due to a still wide enough spread.

Specifically, looking at the charge variable results in both models, the basic network shows the propagation of intervals both upward and downward relatively equally. As mentioned, this is not possible as it indicates a reversal of the battery's degradation. On the contrary, the complex network's charge variables shows both prediction intervals decreasing after 8000 cycles. This indicates that both the model has learned and is far more accurate than the basic model.

6.2.4 General Comparison

In general, the complex network was able to more coherently and confidently quantify the uncertainty of the system. This is believed to be as a result of the stabilisation added by the voltage and charging time variables which behave more predictably than the maximum charging temperature by itself. Due to the stability added, the predictive model is more confident in the derived relationship and makes more accurate forward predictions in the long term. Despite the lack of stability of the temperature variable, its inclusion in the network is beneficial when used in combination with other variables. In the univariate case, it inflates uncertainty due to the inherent noise that exists with the variable. However, in the multivariate case, it acts as a variable which adds context or helps resolve ambiguity due to its high mutual information.

6.3 Future Work

Despite increasing adoption of data-driven methods in health diagnostics, their use in long-horizon degradation modeling and RUL estimation remains sparse and underdeveloped — particularly in systems where measurement sparsity and nonlinear degradation are coupled. This paper focuses itself on RUL prediction and predictive maintenance modelling in a complex system of interacting variables which remains an area with expansive potential studies. This section is devoted to the possible future explorations into this area.

6.3.1 Future Applications

The best applications of DBNs are those where a known relationship exists between variables but due to noisy or infrequent measurements, there exists uncertainty in their relationship with each other. It may also be that in the context of a system, the relationship may be so complex that traditional methods can not cope with the

high dimensionality of the system.

Short Term Predictive Modelling

For systems where real-time state estimation of a variable is required, a short term model that prioritises responsiveness is ideal. Extending the RUL of the battery idea, this may be to monitor and optimise charging temperature that has the least negative impact on the battery's health. For these short term applications, a method of inference that is not computationally complex but still makes informed decisions using all available information is required. For this reason, a coupled model that uses a hybrid physics based knowledge of the system in combination with a SMCM could be used to optimise responsiveness and provide the most informed information leading to a measured charging temperature. These applications would be especially effective when high-resolution measurements are available intermittently, enabling data-efficient belief correction.

Similar applications to this may be fault detection in machinery whereby a hidden variable that may be informed from a system of measurables. By creating a DBN and using a correct inference method, systematic checks of the machinery could be made at optimal times to balance risk of failure and workflow efficiency. In this case, responsiveness of the predictive model may be granted leniency relative to that of fault diagnostics where risk of failure is more critical.

Long Term Predictive Modelling

For systems that require a vision that extends far into the future, the main focus should be on the transition dynamics from one prediction to the next and inferring a confident prediction of the future and the prediction range. Due to the potentially slow moving dynamics of the system where low-frequency and irregular measurements are available, the accuracy takes priority over the responsiveness and uncertainty

quantification is critical.

As mentioned previously, an inner and outer DBN could be used in a case like this where long periods of time may go by without a measurement update and then a burst of data will flood the system. This mismatch in temporal granularity and data richness would require a two level, hierarchical DBN where short interval data could be used to make long term forward predictions. This is certainly an area to be explored for the case of the batteries as a short term burst of temperature data across one cycle could help to inform the state of health of the battery across the next one hundred cycles.

Large Scale Applications

The idea of a hierarchical DBN extends to applications where diagnostics of a full fleet of systems could be used to analyse trends across a full population of systems and track specific hidden variables that might exist within a singular system. This hierarchical DBN would also enable parameter sharing between systems. These models could support transfer learning across devices, where shared structures inform initial priors and device-specific components adapt via local inference. The main issue would be a large computational expense which would require a well-designed method of inference, especially if the population of systems was in the thousands or millions. This idea could be extended to fleets of wind turbines, electric vehicles, and many more large scale, data rich applications.

6.3.2 Transition Models

The future work may be split into the numerous potential applications but also within the method lies future areas of study. Within each potential method, lies problems that are studied throughout statistics and data science, for example, the adaptive grid method that formed my transition model, lies many potential improvements including

local kernels and bandwidths within each local cluster to better reduce granularity and increase resolution of the multi-modality featured in the areas of interest within the larger data set.

However, transition models don't always have to work alone and can be merged with other models. This section is devoted to the types of transition models and the potential of each in combination with others to be used in this specific application and in others.

Physics Based Models

Physics based models are derived from underlying knowledge of the mechanics of a system. In this case, a physics based model could be derived from electro-chemistry physics to better transition particles. This would create multi-modal distributions that can better represent the possible states of a system by incorporating the knowledge of the system and adding stochastic noise and processes to represent the possible future states of the system. This is discussed in section 2.2.3, with the reference to (Z. Li et al. 2011), where a fracture mechanics transition model was used. Future work might use either a data-driven auxiliary transition model and incorporate an underlying knowledge of the system into the particle dynamics.

Parametric Data-Driven Models

Data-driven models may be split into two families - parametric and non-parametric, similar to Bayes filters. A parametric transition model assumes the probability distribution from which the particles are sampled is from a fixed family of functions which are described by a finite set of parameters. Commonly used examples are linear models, certain regression models, but less known, and often assumed non-parametric, are neural network transition models. Parametric models are generally less computationally expensive than non-parametric models but can lack flexibility. Relating

these methods to applications, short term, small scale applications would better suit these methods due to their responsiveness. Further study into the exact pairing of application and model would see this and would warrant studies.

The exception to this generalisation is neural networks, which are often assumed non-parametric as they can consist of thousands of weights and are, therefore, highly-flexible but still consist of a fixed architecture of weights. These weights act as parameters, and so therefore, neural network transition models are parametric models but have the characteristics of a non-parametric method i.e., highly flexible and computationally expensive. A study into their suitability as transition models in real world applications is an area of high interest and potential.

Non-Parametric Data-Driven Models

The transition model of KDE used in this thesis is a non-parametric model due to its number of parameters being based on the number of samples available i.e., no fixed model architecture. As one might gather from this thesis, and discussed in the previous section, it is computationally expensive relative to parametric methods but is highly adaptable. This suits applications where the system is highly complex where fixed parameter models can not be fitted to the system easily. Other non-parametric methods exist, most notably, Gaussian processes, random forest regression and many others. These transition models suit applications where a large computational expense can be devoted to the application if the responsiveness is not a priority i.e., large scale and long term applications.

Chapter 7

Conclusion

In this thesis, two data-driven approaches were presented for predictive modeling of battery degradation using dynamic bayesian networks, each implemented with particle filtering to enable sequential inference under uncertainty. The primary goal was to infer the state of hidden degradation-related variables from noisy and intermittent observations, and to then extend these hidden states forward in time to predict the long-term health trajectory of the battery. These methods were designed to operate effectively under the real-world conditions of sparse measurements, process noise, and the nonlinear evolution of battery behavior across thousands of charge-discharge cycles.

The first approach, detailed in Chapter 4, introduced a two-node DBN composed of maximum charging temperature and maximum state of charge. While the model succeeded in performing belief updates following sparse observations, it lacked long-term coherence. Over extended time horizons, the particle filter failed to converge toward a meaningful posterior distribution, and the model exhibited unstable, often divergent behavior in its predictions. The belief state did not stabilize, and the results were inconsistent across repeated runs. These findings highlight the limitations of using a minimal variable set and underscore the importance of variable selection and

model complexity in stabilizing predictive inference.

The second approach, described in Chapter 5, introduced a more complex DBN with four observable nodes: charge time, maximum voltage, maximum charge, and maximum charging temperature. This model significantly improved performance in both short-term responsiveness and long-term degradation tracking. The inclusion of additional variables, even those with higher individual noise, contributed to greater joint informativeness and more stable belief updates. Notably, variables like charge time, which on their own offer low predictive reliability, served to regularize belief evolution when conditioned on other features, contributing to the network’s stability and expressiveness.

One of the most significant outcomes was the complex model’s ability to learn and generalize the long-term degradation trend, particularly in projecting the gradual decline of battery capacity over hundreds of cycles. The model’s predictions remained coherent even when extended well beyond the final measurement update, with tight posterior distributions that reflected both learned trends and uncertainty propagation. Crucially, this was achieved by the use of the adaptive KDE to localize evaluation effort in high-density regions of the belief state, enabling more efficient and focused inference.

To assess the robustness of belief propagation, the ESS of the particle filter was tracked over time. In the four-node model, the ESS remained relatively high, even in the absence of regular observations. This indicated that the filter maintained particle diversity and avoided degeneracy — a clear sign that the model had converged to a meaningful and stable belief distribution.

Together, these two approaches illustrate the trade-off between architectural simplicity and inference quality. While the simpler model offered responsiveness and ease of implementation, the complex network provided superior stability, coherence, and predictive depth — particularly valuable for applications involving long-term

degradation forecasting.

Looking ahead, several extensions present themselves. Hierarchical or nested DBNs could be used to reconcile irregular measurement regimes, where sparse, low-frequency observations are occasionally supplemented by high-resolution bursts of data. Fleet-wide DBNs could also be explored, where parameter sharing and population-level trends enhance the predictive power for individual systems. With continued attention to inference scalability, transition model design, and real-world deployment, DBNs hold strong potential as the backbone of intelligent, uncertainty-aware monitoring frameworks for energy storage and beyond.

In conclusion, this work offers a flexible, interpretable, and scalable framework for predictive maintenance, but its relevance extends beyond maintenance alone — providing a foundation for the design, optimization, and control of intelligent systems where uncertainty, degradation, and sparse observations must be robustly managed. By combining probabilistic reasoning, adaptive inference, and domain-informed modeling, it brings predictive modeling a step closer to practical deployment in critical, real-world systems.

Part II

Appendices

Appendix A

Figures

Particle Paths

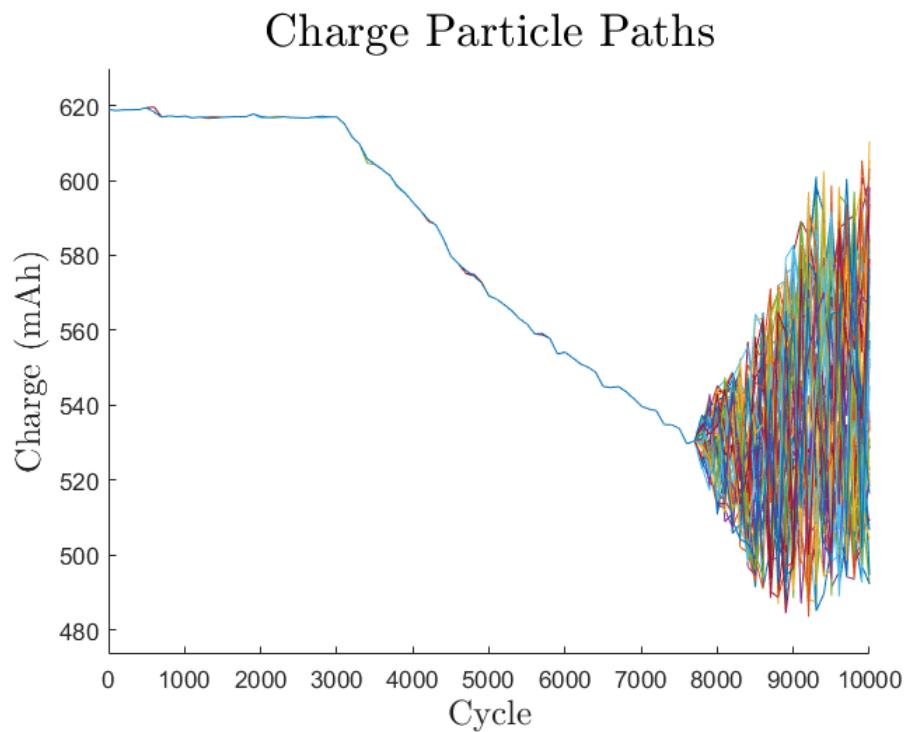


Figure A.1: Particle paths for charge variable in the basic network.

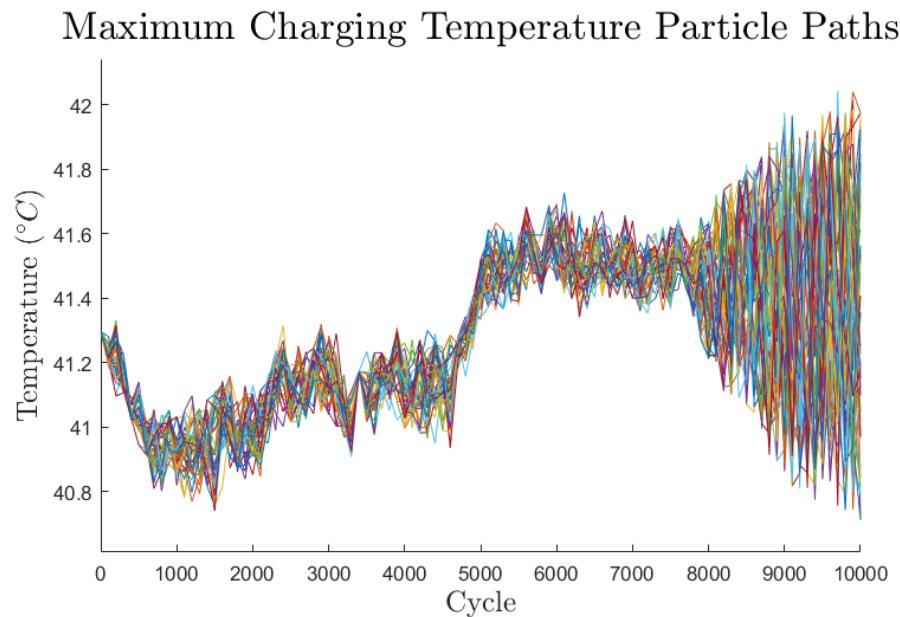


Figure A.2: Particle paths for maximum charging temperature variable in the basic network.

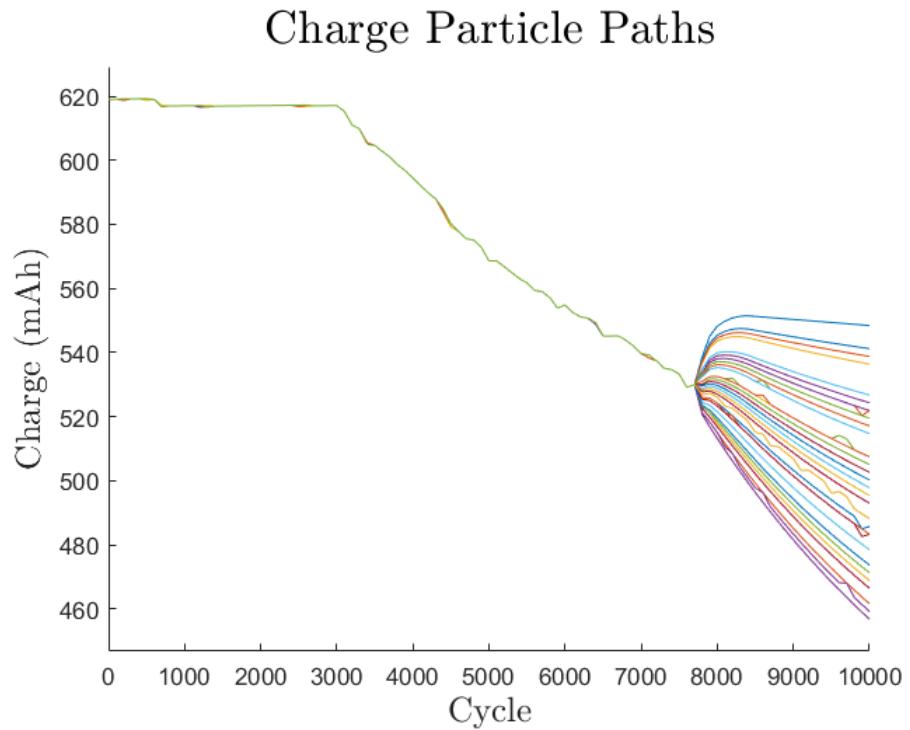


Figure A.3: Particle paths for charge variable in the complex network.

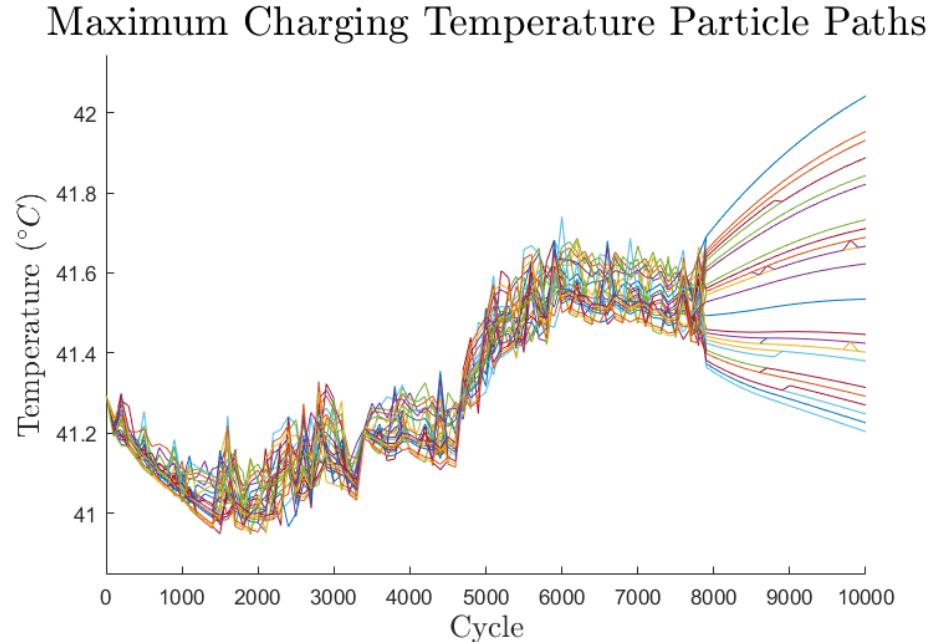


Figure A.4: Particle paths for maximum charging temperature variable in the complex network.

Maximum Charging Voltage Particle Paths

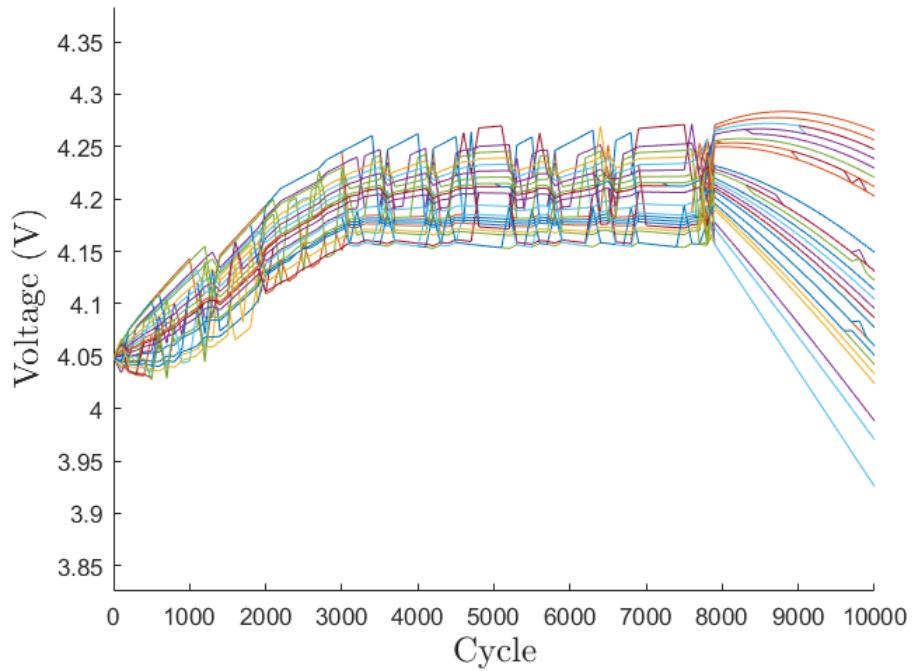


Figure A.5: Particle paths for maximum charging voltage variable in the complex network.

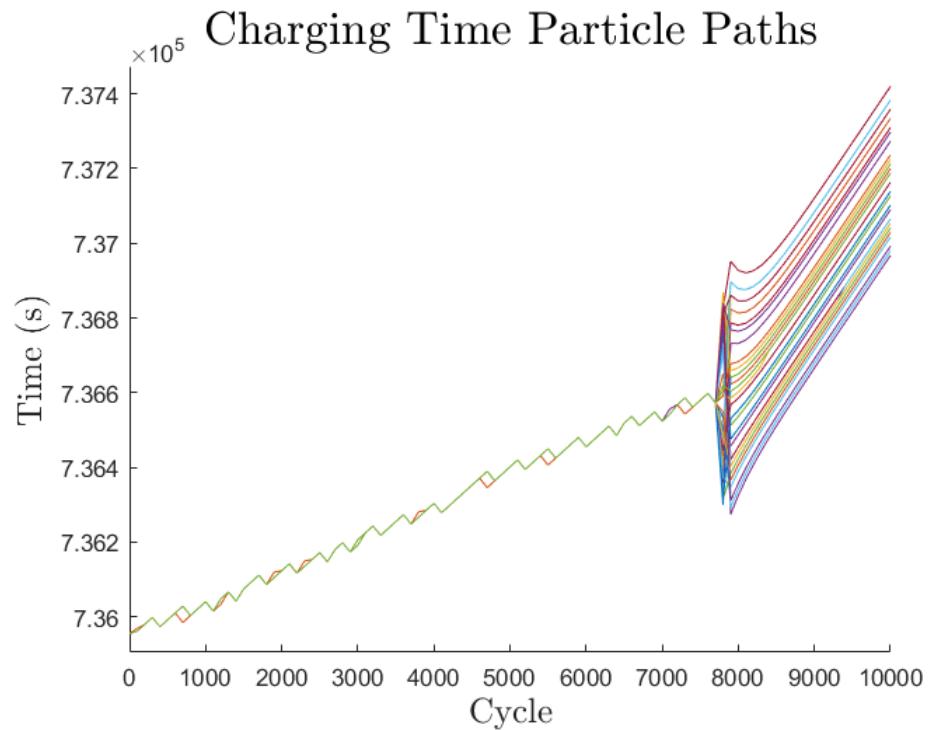


Figure A.6: Particle paths for charging time variable in the complex network.

Bibliography

- Arsenault, Ryan (Apr. 2016). *The Rising Cost of Downtime*. Aberdeen Research Group.
- Assembly, United Nations General (2015). *Transforming Our World: The 2030 Agenda for Sustainable Development*.
- Baum, Leonard E and Ted Petrie (1966). “Statistical inference for probabilistic functions of finite state Markov chains”. In: *The annals of mathematical statistics* 37.6, pp. 1554–1563.
- Bellman, Richard, Robert E Kalaba, et al. (1965). *Dynamic programming and modern control theory*. Vol. 81. Citeseer.
- Birkl, Christoph (2017). “Diagnosis and prognosis of degradation in lithium-ion batteries”. PhD thesis. University of Oxford.
- Bowman, Adrian W (1984). “An alternative method of cross-validation for the smoothing of density estimates”. In: *Biometrika* 71.2, pp. 353–360.
- Cappé, O., E. Moulines, and T. Rydén (2005). *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer, pp. 1–5. ISBN: 9780387402642.
- Chacón, José E and Tarn Duong (2018). *Multivariate kernel smoothing and its applications*. Chapman and Hall/CRC.
- Chen, Di (2012). “Microscopic Investigations of Degradation in Lithium-Ion Batteries”. 43.15.02; LK 01. PhD thesis. Karlsruher Institut für Technologie (KIT).

- Chen, Serena H and Carmel A Pollino (2012). “Good practice in Bayesian network modelling”. In: *Environmental Modelling & Software* 37, pp. 134–145.
- Darányi, András, Tamás Ruppert, and János Abonyi (2024). “Particle filtering supported probability density estimation of mobility patterns”. In: *Heliyon* 10.8.
- Glaessgen, Edward and David Stargel (2012). “The digital twin paradigm for future NASA and US Air Force vehicles”. In: *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, p. 1818.
- Hall, Peter et al. (1991). “On optimal data-based bandwidth selection in kernel density estimation”. In: *Biometrika* 78.2, pp. 263–269.
- MIT Course Catalog Bulletin 2024-2025* (n.d.).
- Jones, MC and SJ Sheather (1991). “Using non-stochastic terms to advantage in kernel-based estimation of integrated squared density derivatives”. In: *Statistics & Probability Letters* 11.6, pp. 511–514.
- Jurafsky, Daniel (2000). *Speech and language processing*.
- Kalman, Rudolph Emil (1960). “A new approach to linear filtering and prediction problems”. In: *Journal of Basic Engineering*.
- Khodarahmi, Masoud and Vafa Maihami (2023). “A review on Kalman filter models”. In: *Archives of Computational Methods in Engineering* 30.1, pp. 727–747.
- Kjaerulff, UB (2008). *Bayesian networks and influence diagrams*.
- Ko, Nak Yong and Tae Gyun Kim (2012). “Comparison of Kalman filter and particle filter used for localization of an underwater vehicle”. In: *2012 9th international conference on ubiquitous robots and ambient intelligence (URAI)*. IEEE, pp. 350–352.
- Kong, Augustine (1992). “A note on importance sampling using standardized weights”. In: *University of Chicago, Dept. of Statistics, Tech. Rep* 348, p. 14.

- Lemke, Dorothea et al. (2015). “Comparing adaptive and fixed bandwidth-based kernel density estimates in spatial cancer epidemiology”. In: *International journal of health geographics* 14, pp. 1–10.
- Li, Chenzhao et al. (2017). “Dynamic Bayesian network for aircraft wing health monitoring digital twin”. In: *Aiaa Journal* 55.3, pp. 930–941.
- Li, Zhe et al. (2011). “Modeling the capacity degradation of LiFePO₄/graphite batteries based on stress coupling analysis”. In: *Journal of Power Sources* 196.22, pp. 9757–9766.
- Lucas, Peter JF, Linda C Van der Gaag, and Ameen Abu-Hanna (2004). *Bayesian networks in biomedicine and health-care*.
- Market.US (Feb. 2023). “Lithium Ion Battery Market is Slated to be Worth USD 307.8 Billion by 2032”. In: *GlobalNewsWire*.
- Menye, Joselyn Stephane, Mamadou-Bailo Camara, and Brayima Dakyo (2025). “Lithium Battery Degradation and Failure Mechanisms: A State-of-the-Art Review”. In: *Energies* 18.2, p. 342.
- Monroe, Charles and John Newman (2003). “Dendrite growth in lithium/polymer systems: A propagation model for liquid electrolytes under galvanostatic conditions”. In: *Journal of The Electrochemical Society* 150.10, A1377.
- Musso, Christian, Nadia Oudjane, and Francois Le Gland (2001). “Improving regularised particle filters”. In: *Sequential Monte Carlo methods in practice*. Springer, pp. 247–271.
- Ohsaki, Takahisa et al. (2005). “Overcharge reaction of lithium-ion batteries”. In: *Journal of power sources* 146.1-2, pp. 97–100.
- Parzen, Emanuel (1962). “On estimation of a probability density function and mode”. In: *The annals of mathematical statistics* 33.3, pp. 1065–1076.

- Rastelli, Riccardo, Florian Maire, and Nial Friel (2024). “Computationally efficient inference for latent position network models”. In: *Electronic Journal of Statistics* 18.1, pp. 2531–2570.
- Rosenblatt, Murray (1956). “Remarks on Some Nonparametric Estimates of a Density Function”. In: *The Annals of Mathematical Statistics* 27.3, pp. 832–837. DOI: 10.1214/aoms/1177728190. URL: <https://doi.org/10.1214/aoms/1177728190>.
- Rudemo, Mats (1982). “Empirical choice of histograms and kernel density estimators”. In: *Scandinavian Journal of Statistics*, pp. 65–78.
- Russell, Stuart J and Peter Norvig (2016). *Artificial intelligence: a modern approach*. Pearson.
- Särkkä, Simo and Lennart Svensson (2023). *Bayesian filtering and smoothing*. Vol. 17. Cambridge university press.
- Sheather, Simon (1983). “A data-based algorithm for choosing the window width when estimating the density at a point”. In: *Computational Statistics & Data Analysis* 1, pp. 229–238.
- Sheather, Simon J (1986). “An improved data-based algorithm for choosing the window width when estimating the density at a point”. In: *Computational Statistics & Data Analysis* 4.1, pp. 61–65.
- Silverman, Bernard W (1986). *Density estimation for statistics and data analysis*. Routledge.
- Sinaga, Kristina P. and Miin-Shen Yang (2020). “Unsupervised K-Means Clustering Algorithm”. In: *IEEE Access* 8, pp. 80716–80727. DOI: 10.1109/ACCESS.2020.2988796.
- Thrun, Sebastian (2002). *Probabilistic robotics*. Vol. 45. 3. ACM New York, NY, USA, pp. 52–57.
- Turlach, Berwin (Feb. 1999). “Bandwidth Selection in Kernel Density Estimation: A Review”. In: *Technical Report*.

- Wu, Yali et al. (June 2017). “A self-adaptive chaos and Kalman filter-based particle swarm optimization for economic dispatch problem”. In: *Soft Computing* 21. doi: 10.1007/s00500-015-2013-x.
- Zhang, Jingyi et al. (2023). “An optimal transport approach for selecting a representative subsample with application in efficient kernel density estimation”. In: *Journal of Computational and Graphical Statistics* 32.1, pp. 329–339.